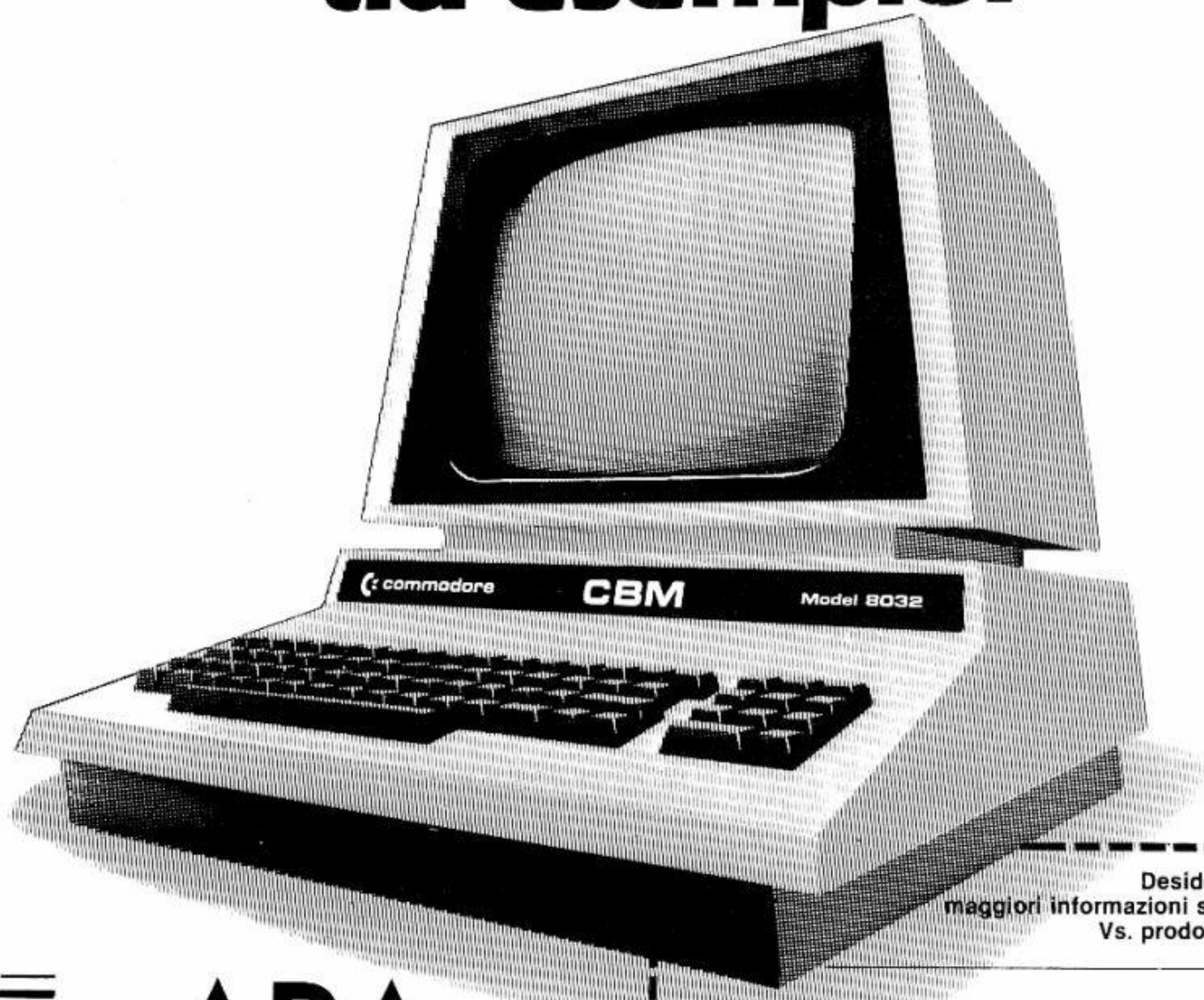
commodore
COMPUTER
CLUB
N. 4
Lire 2000
La rivista degli utenti di sistemi Commodore
25 Maggio 1983 - Sped. in abb. post. gr. III/70 - Suppl. a Computer n. 58 - Tassa pagata per IP
I bioritmi delle elezioni
Il software del '64
Programmi per il Vic
e i modelli CBM
1
2
3
4

ABA: la microinformatica, chiavi in mano.

ABA ELETTRONICA non si limita a trattare la più ampia gamma di marche e di modelli per tutte le applicazioni, da quelle hobbistiche alle gestionali. ABA ELETTRONICA mette a vostra disposizione il mondo della microinformatica, dai corsi di istruzione a vari livelli, all'assistenza tecnica più qualificata, alla vendita di periferiche, accessori e pubblicazioni. Vi aiuta a scegliere inoltre. Nella sua sala di dimostrazione è possibile provare e confrontare quanto di meglio offre oggi il mercato. E quando avrete deciso per un microcomputer, ABA ELETTRONICA vi propone di scegliere la forma di acquisto che preferite. Anche in leasing o per corrispondenza. Infine ABA ELETTRONICA vi fornisce tutti i programmi, standard o su misura, siano essi gestionali, professionali o scientifici che Vi necessitano provvedendo anche all'addestramento dell'operatore sul sistema che avete scelto e su tutta la microinformatica che lo riguarda. Chiavi in mano.

Vendita, Programmazione e Assistenza:
ABA ELETTRONICA - 10141 Torino - Via Fossati 5/c
Tel. (011) 33.20.65/38.93.28

SOMMARIO

Commodore Computer Club - Rivista indipendente per gli utenti di sistemi Commodore.
Direttore responsabile: Michele di Pisa
Redazione: Alessandro De Simone, Franco Rao
Direzione, redazione: Piazza Arduino, 3 - 20149 Milano - Tel. (02) 434354-435717
Pubblicità: Milano: Paola Bevilacqua, Gianluigi Centurelli, Mario Gandolfo, Tina Ronchetti, Villa Claudio. - Viale Famagosta, 75 - 20142 Milano - tel. (02) 8467348/9/40
Prezzi e abbonamenti: la rivista esce bimestralmente. Prezzo per una copia Lire 2.000. Arretrati il doppio. Abbonamento per dieci fascicoli lire 18.000. Abbonamento annuo cumulativo alle riviste Computer e Commodore Computer Club (tariffa riservata agli studenti): L. 24.000. I versamenti vanno indirizzati a Minisystems Italia s.r.l., mediante assegno bancario, vaglia o utilizzando il c/c postale n. 11909207.
Composizioni: Minisystems Italia
Selezioni: Org. Aldo Ghiacci
Stampa:
Registrazione: Tribunale di Milano n. 370 del 2/10/1982 - Sped. in abb. post. gr. III n. 70 quale supplemento alla rivista Computer - Pubbl. inferiore al 70%

DOMANDE RISPOSTE

Sin dal momento del suo lancio sul mercato ci sono state poste delle domande ricorrenti sul Commodore 64. Le più frequenti sono le seguenti.

Che tipo di BASIC usa?

Il BASIC 2,0, lo stesso del Vic-20.

I programmi PET si possono applicare sul 64?

Esiste un emulatore PET che emula l'originale della vecchia ROM 2001. Pertanto, qualsiasi programma in BASIC 2 che non usi PEEK, POKE, USR, o SYS dovrebbe funzionare.

Il Commodore 64 può essere usato su un televisore domestico?

Sì! E' sufficiente collegarlo alla presa.

Posso imparare a programmare con il Commodore 64?

Sì! La fortunata serie "Introduzione al BASIC" è stata adattata al Commodore 64 ed è ora disponibile (vedi l'articolo sui programmi disponibili).

Il software del Vic-20 è applicabile al Commodore 64?

Qualsiasi programma che non usi PEEK, POKE, USR o SYS dovrebbe funzionare. I programmi che contengono questi comandi dovranno essere modificati per una diversa configurazione di memoria del 64. I programmi su cassetta del Vic-20 non possono essere direttamente caricati sul 64. Questo è dovuto alla diversa velocità di trasferimento dei dati delle due macchine. Questi programmi devono prima essere trasferiti dalla cassetta al dischetto e quindi caricati.

Le cartucce del Vic funzionano sul 64?

No perchè la configurazione di memoria del 64 è diversa da quella del Vic-20, le cartucce giochi per il Vic-20 non possono essere usate sul 64.

Che cosa sono gli Sprites?

Si tratta di blocchetti programmabili che si possono muovere sullo schermo. Sino ad otto sprites indipendenti si possono spostare contemporaneamente.

Esiste un codice monitor integrato?

No. Comunque un sistema di sviluppo assembler sarà disponibile all'inizio di aprile.

MEE
Tecnologia del domani
flexible disks
minidisks
Verbatim
Datalife
Datalife
Verbatim
Verbatim
Verbatim
Verbatim
minidisk
Verbatim
Head Cleaning Kit
Datalife
Verbatim
Head Cleaning Kit
Alignment minidisks
Alignment Diskette
Verbatim
Alignment Diskette Optima Series
Verbatim
Verbatim
La MEE esclusivista per l'Italia dei prodotti VERBATIM DATALIFE Vi propone oltre alla più completa gamma di supporti magnetici anche armadi ignifughi Rosengrens, nastri inchiostrati per tutte le stampanti, mobili e tavoli porta terminali per l'arredamento dei CED.
Con i prodotti MEE le registrazioni dei Vostri dati non hanno più problemi di affidabilità.
MEE - Memorie per Elaboratori Elettronici s.r.l.
Forniture per Centri Elaborazione Dati
Sede Amm.va: 20144 Milano - Via Boni, 29
Tel. 4988541 (4 linee r.a.) 4986296-4984196
Filiali e Agenzie: Milano - Bergamo - Torino - Biella - Padova - Parma - Bologna - Firenze - Ancona - Roma - Napoli - Catania - Oristano - Bari - Genova - Bolzano - Mestre -
Vi prego vol
inviare la vost
comple
documentazio
relativa
Dische
e inolt
Disk-packs
Nastri inchiostr
Armadi ignifughi
Mobili per CE
ragione sociale
nome
via
Cap.
Città

VIC 20 un vero sistema

Cassette per espansione di memoria.
VIC 20 ha una memoria di base di 5 K bytes. Per aumentarla sino a 32 K bytes hai a disposizione tre cassette da 3, 8 e 16 K RAM. Basta inserirle nel VIC oppure nell'apposito Memory expansion board.

VIC 20 è un sistema interamente espandibile grazie alla semplice aggiunta di moduli, questi che vedi e molti altri.

VIC 20 Computer.
Questo è il favoloso VIC 20 computer. 5 K-Bytes espandibili a 32. 24 colori. Note musicali. Collegabile all'unità nastro, all'unità disco e alla stampante. Collegabile in Modem con le normali linee telefoniche.

VIC 20 Single drive floppy disk. Unità a disco.
Questo "floppy" consente di sfruttare tutta la potenzialitá del VIC 20. Mette a disposizione un metodo veloce ed efficiente per la memorizzazione e il recupero dei dati e dei programmi.
Abbinando il floppy disk alla stampante, il VIC 20 diventa un computer system ideale per il piccolo imprenditore, per il professionista, per le ricerche.

Cassette Unit. Unità a nastro.
È questo il primo degli accessori del VIC. Viene utilizzata per memorizzare dati e programmi o per inserire cassette con programmi già pronti.
Si collega direttamente con il VIC 20.

Stampante VIC 20.
Come ogni periferica VIC 20 questa stampante è prodotta con alta specializzazione e basso costo.
Ha mille utilizzi.
Per fare copie dei programmi, per fare lettere circolari, grafici, dati di lavoro, etc. Per realizzare programmi di word processing.

Commodore delle mie brame, chi è il più bello del reame?

In base al bioritmo disegnato da un Commodore 64, alcuni leader politici arriveranno alle votazioni in condizioni psico-fisiche piuttosto disastrate. Se lo avessero saputo prima, la data delle elezioni sarebbe stata certamente diversa.

QUANTO saranno in forma i nostri uomini politici durante il periodo elettorale? Un Commodore 64 risponde a questa domanda tracciando il bioritmo di quattro personaggi di spicco: Berlinguer, Craxi, De Mita e Pannella.

Per quanti non lo sappiano ancora (pochissimi fans dei personal computer, crediamo) ricordiamo che secondo la teoria dei bioritmi il corpo umano ha degli orologi interni o ritmi metabolici a cicli di tempo costanti.

Si pensa che esistano tre cicli che iniziano alla data di nascita in direzione positiva.

Il ciclo di 23 giorni, o fisico, riguarda: la validità, la resistenza fisica e l'energia.

Il ciclo di 28 giorni, o sensitivo, riguarda la sensibilità, l'intuizione e il buonumore.

Il ciclo di 33 giorni, o conoscitivo, riguarda la prontezza mentale e il discernimento.

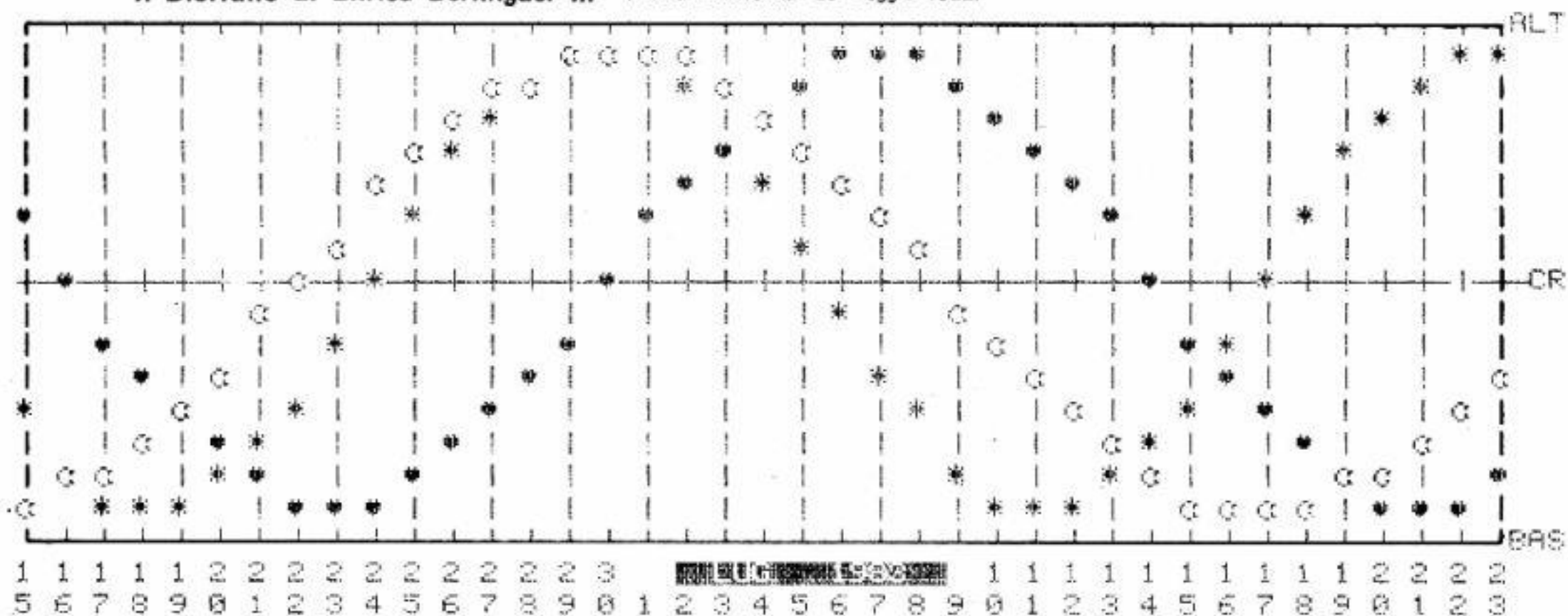
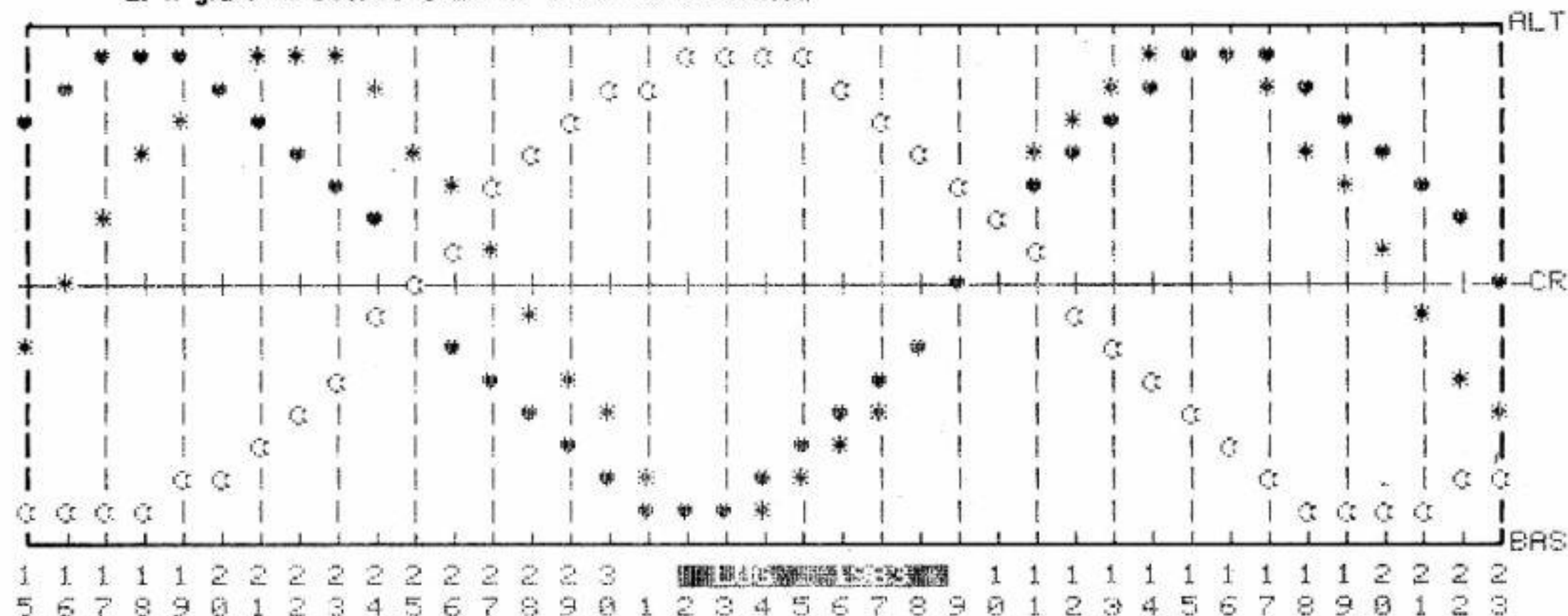
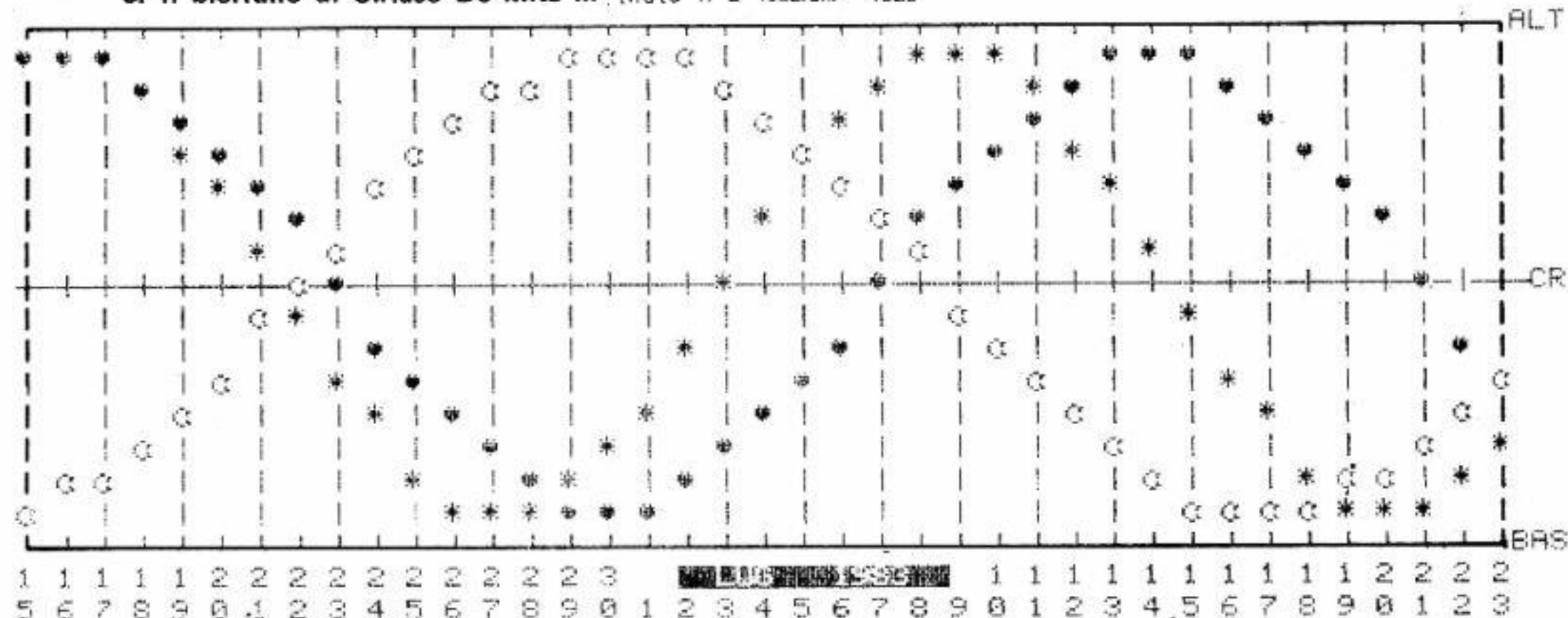
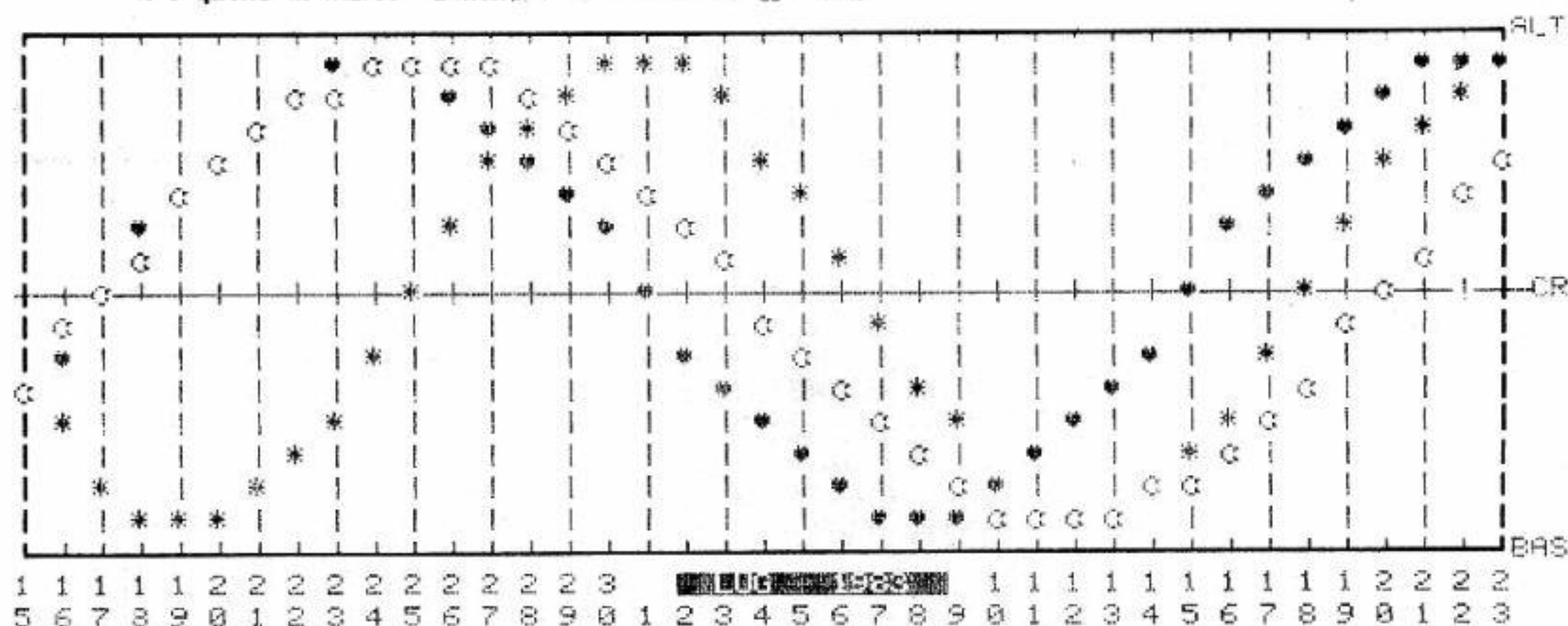
Per ogni ciclo, un giorno può essere considerato alto, basso o critico.

Nei periodi alti (0<X<=1), i più energici, ci si sente più dinamici. Durante i periodi bassi (-1<=X<0), si recupera l'energia spesa durante il periodo attivo.

Nei periodi critici (X = 0) specialmente ai "fisici" ed ai sensitivi, possono accadere incidenti.

I programmi di bioritmo sono disponibili su nastro per qualsiasi modello di computer CBM. Molti utenti, però, preferiscono scrivere da sè il proprio bio-programma. Ecco, comunque, il listato pervenutoci da un lettore.

```
100 REM BIORITMI
110 REM COPYRIGHT BY IVANO BIFFI 1982
120 L=0:Z=.9999:T=6:P=3.14159265
130 PRINTCHR$(147);TAB(14);"BIORITMI"
140 GOSUB1200:PRINT:PRINT
150 PRINT"DATA DI NASCITA"
160 GOSUB500
170 GOSUB600
180 JB=JD
190 PRINT:PRINT"DA CHE GIORNO ?"
200 GOSUB500:GOSUB1231
210 GOSUB600
220 JC=JD
230 IFJC>=JBTHEN270
260 GOTO140
270 FORJ=1TO1000:NEXT
280 GOSUB1230:GOSUB700
300 N=JC-JB
310 V=23:GOSUB800:GOSUB850
320 V=28:GOSUB800:GOSUB850
330 V=33:GOSUB800:GOSUB850
340 GOSUB1000
350 PRINTC$;TAB(8);L$
360 JC=JC+1:L=L+1:IFL<15THEN300
370 PRINT:PRINT"'F'FINE,'SPACE'CONTINUA"
380 GETR$:IFR$=""THEN380
390 IFR$="F"THEN120
400 L=0:GOTO280
500 PRINT
505 INPUT"GIORNO(DA1A31)";D
506 IFD=0THEN1235
510 D=INT(D):IFD<1ORD>31THEN505
520 INPUT"MESE(DA1A12)";M
530 M=INT(M):IFM<1ORM>12THEN520
540 INPUT"ANNO";Y
550 Y=INT(Y):IFY<0THEN540
560 IFY>99THEN580
570 Y=Y:PRINT"VA BENE,";Y
580 RETURN
```

1. Bioritmo di Enrico Berlinguer ... (data di nascita: 25 maggio 1922)
ALT
CR
BAS
2. Il grafo di Bettino Craxi ... (nato il 24 febbraio 1934)
ALT
CR
BAS
3. Il bioritmo di Ciriaco De Mita ... (nato il 2 febbraio 1926
ALT
CR
BAS
* = CONDIZIONE FISICA
♥ = STATO EMOTIVO
C = LIVELLO INTELLETTUALE

4. e quello di Marco Pannella (data di nascita: 2 maggio 1930)

```
* = C O N D I Z I O N E   F I S I C A
♥ = S T A T O   E M O T I V O
ⵛ = L I V E L L O   I N T E L L E T T U A L E
```

```
600 W=INT((M-14)/12+Z)
610 JD=INT(1461*(Y+4800+W)/4)
620 B=367*(M-2-W*12)/12
630 IFB<0THENB=B+Z
640 B=INT(B):JD=JD+B
650 B=INT(INT(3*(Y+4900+W)/100)/4)
660 JD=JD+D-32075-B
670 RETURN
700 PRINTCHR$(147);
710 PRINTTAB(14);"BIORITMI"
720 PRINT:PRINT"DATA";TAB(0);
730 PRINT"     NEG";TAB(8);"    0";"    POS"
740 PRINTTAB(8);U$
750 RETURN
800 W=INT(N/V):R=N-W*V
810 RETURN
850 IFV<>23THEN900
855 L$=CHR$(32):FORJ=1TO3
860 L$=L$+L$:NEXT
870 L$=L$+LEFT$(L$,6)
880 L$=LEFT$(L$,T)+CHR$(194)+RIGHT$(L$,T)
890 IFV=23THENC$="F"
900 IFV=28THENC$="E"
910 IFV=33THENC$="I"
920 W=R/V:W=W*2*P
930 W=T*SIN(W):W=W+T+1.5
940 W=INT(W):A$=MID$(L$,W,1)
950 IFA$="F"ORA$="E"ORA$="*"THENC$="*"
955 IFW=1THEN980
957 IFW=31THEN990
960 L$=LEFT$(L$,W-1)+C$+RIGHT$(L$,13-W)
970 RETURN
980 L$=C$+RIGHT$(L$,12):RETURN
990 L$=LEFT$(L$,12)+C$:RETURN
1000 W=JC+68569:R=INT(4*W/146097)
1010 W=W-INT((146097*R+3)/4)
1020 Y=INT(4000*(W+1)/1461001)
1030 W=W-INT(1461*Y/4)+32
1040 M=INT(80*W/2447)
1050 D=W-INT(2447*M/80)
1060 W=INT(M/11):M=M+2-12*W
1070 Y=100*(R-49)+Y+W
1080 A$=STR$(D):W=LEN(A$)-1
1090 C$=MID$(A$,W,2)+"/"
```

```
1100 A$=STR$(M):W=LEN(A$)-1
1110 C$=C$+MID$(A$,W,2)+"/"
1120 A$=STR$(Y):W=LEN(A$)-1
1130 C$=C$+MID$(A$,2,W)
1140 RETURN
1200 U$=CHR$(32):U$=CHR$(210):FORJ=1TO3
1210 U$=U$+U$:NEXT
1220 U$=U$+LEFT$(U$,6)
1230 RETURN
1231 POKE36878,10
1232 FORL=1TO100
1233 POKE36876,INT(RND(1)*128)+128
1234 FORX=1TO10
1235 NEXTX:NEXTL
1236 POKE36876,0:POKE36878,0
1237 RETURN
```

Distributori

Computer Market all'ombra del Vesuvio

La Computer Market è stata costituita nel febbraio 1983 su iniziativa di alcuni imprenditori napoletani, già inseriti nel mondo EDP, desiderosi di proporsi all'attenzione di un mercato in grosso fermento come quello della Campania del post-terremoto con un prodotto che costituirà il grosso business degli anni '80.

La Computer Market, concessionaria regionale della Commodore, si pone degli obiettivi molto ambiziosi il cui raggiungimento ha già da ora imposto certe scelte molto precise e definite. Una di queste scelte riguarda la casa mandante. Infatti la Computer Market opererà solo con prodotti Commodore sia già presenti sul mercato italiano sia di prossimo annuncio.

A livello obiettivi, uno dei primi da raggiungere è quello di rendere la giusta immagine al nome Commodore. Bisogna convenire sul fatto che prima del febbraio '83 non esistendo in Campania un Dealer vero e proprio le vendite venivano effettuate alla "buona" da chi poteva. Ovvero da distributori delle regioni limitrofe, da qualche rivenditore locale, magari con una struttura ancora inadeguata, ma principalmente da un "sottobosco" di persone improvvisatesi venditori a tempo parziale e con strutture pressappoco inesistenti, con listini incontrollati vendevano sfruttando un po' le conoscenze ed un po' la diffusione del nome Commodore. Oggi in Campania esistono alcuni scontenti in primo luogo per la mancata gestione di quell'aspetto fondamentale che è il post-vendita.

Altro obiettivo, anche esso di primaria importanza, è quello di creare per tutti i rivenditori un reale centro di supporto per tutti i tipi di problematiche. Da quelle tipicamente orientate al marketing (analisi del mercato, ricerca e sviluppo di nuovi canali, studio della concorrenza) a quelle di tipo softwaristico di base ed applicativo, (corsi di formazione specialisti, diffusione di nuovi pacchetti applicativi, divulgazione del word-processing e del data-base ecc.). Da problematiche connesse alla fornitura di supporti magnetici e cartacei al Site-preparation globale.

Anche alla manutenzione la Computer Market sta dedicando molte energie per l'allestimento di un laboratorio di manutenzione e riparazione fornito di tutte le parti di ricambio, di tutte le attrezzature di diagnosi e prova compreso il logic fault locator e soprattutto di tecnici di alta qualificazione pronti ad intervenire ed a fornire consigli, strumenti e parti ai tecnici di manutenzione dei rivenditori periferici lontani dal Centro di assistenza di Napoli.

Quando Commodore non fa il crumiro

IL SINDACATO non è rimasto indifferente all'evoluzione tecnologica e per la prima volta si avvicina a livello nazionale al mondo dell'informatica. E' questo il tema trattato nel corso di una conferenza stampa a livello nazionale dal titolo appunto: "La Fib/Cisl adotta il personal computer: un sistema informativo a misura di sindacato" e che si è svolta simultaneamente il 3 maggio in 5 città d'Italia: Roma, Milano, Torino, Udine e Bari.

La Federazione Italiana Bancari (FIB) Cisl, il sindacato di categoria dei lavoratori del settore del credito, ha attivato da circa un anno sul territorio nazionale una rete di telecomunicazioni ed elaborazione locale di dati. La rete è stata realizzata mediante l'installazione di circa 50 personal computer Commodore Serie 8000, interconnessi mediante la rete telefonica pubblica.

I personal computer della Commodore, afferma Sergio Ammannati, segretario generale della Fib/Cisl, sono stati scelti in considerazione del buon rapporto prezzo prestazioni, del grado di modularità e del supporto tecnico-organizzativo offerto dalla Corel di Udine.

Fino ad oggi la tendenza alla realizzazione di un sistema informativo distribuito, in aziende aventi una struttura diffusa sul territorio nazionale, ha incontrato spesso come ostacolo un costo assai elevato a causa dell'assenza di apparecchiature idonee a svolgere contemporaneamente tutte le funzioni richieste e con altissimi costi di installazione e di esercizio. Il personal computer rappresenta ora la soluzione ottimale, offrendo versatilità, facilità d'uso e basso costo (il Commodore 8000 costa meno di un milione e 300 mila lire e in configurazione dagli 8 ai 3,5 milioni). Il package usato, l'Ital-Net, è in grado di computerizzare tutte le filiali di una organizzazione e di garantirne un'ottima velocità di trasmissione di dati o di testi fra le sedi.

L'abbinamento hardware-software, realizzato per la rete Fib nelle sedi periferiche in tutte le regioni italiane, risponde perfettamente alle esigenze del sindacato: maggiore efficienza e tempestività nella trasmissione delle notizie e delle informazioni dal centro alla periferia, con dotazione alle strutture periferiche di uno strumento per l'invio di messaggi e per la gestione anagrafico-amministrativa delle sedi locali, spesso carenti di uomini e di mezzi.

La scelta dei personal computer da parte della Fib/Cisl persegue una strategia di decentramento operativo e culturale che indurrà tutte le sedi periferiche a razionalizzare e qualificare le proprie metodologie di lavoro.

"Una volta per fare il sindacalista era necessario sopratutto un certo 'fiuto'," ha affermato a Milano, il segretario nazionale della Fib/Cisl, Franco Fabrello, "ma oggi questo non basta più, occorrono non solo informazioni aggiornate e immediate ma anche un'elaborazione costante di queste informazioni, sulla base delle quali impostare in modo rigoroso una contrattazione sindacale". ▪

Tutti i package "italiani" del 64

IL COMMODORE 64 è certamente il personal/home computer più versatile che oggi si possa trovare sul mercato. Nessun altro computer, nella stessa gamma di prezzi, è in grado di offrire lo stesso tipo di applicazioni ed una così vasta quantità di programmi software e periferiche. Con una scheda di memoria di 64K, il Commodore 64 può soddisfare qualsiasi richiesta per una vasta gamma di utenti, dalla piccola/-media azienda al più serio ed impegnato programmatore.

Esternamente si presenta molto simile al Vic-20; il Microsoft Basic è la versione 2.0, identica a quello del Vic-20, e ciò ne fa l'evoluzione ideale per i possessori del Vic.

Gli acquirenti del 64, tuttavia, vogliono sapere quale software specifico è già disponibile per questa macchina. Computer Club fa il punto.

Inizia a programmare con il Gortek

Questo piacevole ed unico concetto di insegnare ai giovani come programmare abbina una storia d'avventura spaziale a lezioni di programmazione in BASIC.

Il pianeta Syntax sta per essere invaso dal terribile Zitrous Gortek sta lavorando accanitamente per insegnare ai microchips come programmare il grande computer per affrontare l'attacco nemico. Il libro a colori che illustra questa avventura include il manuale di addestramento dei microchips che insegnerà anche ai bambini i primi rudimenti della programmazione in BASIC.

Il libro che comprende undici belle illustrazioni, a vividi colori è accompagnato da due cassette che contengono giochi istruttivi e altri programmi. Oltre ad usare il programma offerto, andando avanti con il manuale d'istruzione, si trovano programmi che il principiante deve imparare a trasferire sul computer. Alla fine della storia la grande "creatività" del computer è salvata dalla programmazione che il tuo bambino ed i microchips hanno imparato da Gortek. I principianti vincitori si guadagnano il diritto di indossare l'emblema di Gortek, che si trova nel pacchetto.

Questo approccio innovatore per l'insegnamento dell'elaborazione dei dati e la avventura spaziale è il frutto del lavoro di tre insegnanti inglesi. Hanno studiato questo pacchetto appositamente per fanciulli da 10 ai 13 anni anche se può essere raccomandato anche ai più piccoli,

sempre con l'ausilio dei genitori. Ragazzi più grandi e perfino adulti lo troveranno interessante e anche divertente...Si tratta del primo di una serie commercializzata in tutto il mondo esclusivamente dalla Commodore.

Il pacchetto comprende un libro di 44 pagine (38 a colori), due cassette con 12 programmi ed un emblema di Gortek. E' disponibile per Vic-20, Commodore 64 e tutte la Serie 4000.

Un'introduzione al Basic del prof. Andrew Colin

Il popolare pacchetto per autodidatti del Vic 20 è stato revisionato e migliorato dalla versione Vic-20. L'autore, il Prof. Andrew Coli, dell'Università di Strathcyde, dice che "si tratta di un corso imperniato sulla programmazione. Si riferisce al Commodore 64, ma una volta che te ne sei impadronito sarai in grado di usare le tue capacità su qualsiasi altro computer in BASIC, piccolo o grande che sia".

Il corso è diviso in quindici sezioni nella prima parte ed in dieci nella seconda (venticinque sezioni in tutto). In media, per ciascuna sezione dovrai passare una sera o due di lavoro. La maggioranza delle lezioni comprende un po' di esercizio di lettura ed un questionario tipo "auto test", per poter controllare quanto hai capito della lezione stessa. Sono compresi anche degli esperimenti.

Quando si è finito il corso, l'utente è in grado di usare il 64 per molti e svariati modi. Ad esempio, potrà usarlo per tests o quiz, per giochi di propria invenzione.

I giochi o qualsiasi altra applicazione potranno includere disegni a colori e suoni per accentuare quello che intendi, leggiadre note o rumori molesti. Ciascuna delle due parti nel corso comprende due cassette per un totale di 34 programmi per il Commodore 64. Ci sono esempi di programmi, quiz, subroutine che si possono costruire come si vuole.

Attualmente di questo corso è disponibile solo la prima parte (in inglese a lire 25 mila, più Iva). Presto le due parti saranno disponibili in Italiano.

Easy Script

E' uno dei più potenti word processor sul mercato, senza tema di confronto per quanto riguarda il prezzo. I vantaggi del word processing sono ben noti: nessun bisogno di ribattere l'intero documento, possibilità di creare documenti da paragrafi standard, facile creazione di lettere standard personalizzate, possibilità di vedere e di cambiare il documento prima che venga stampato, risparmio di tempo, di spazio e di arrabbiature, e con l'Easy Script sul 64 c'è anche un risparmio di soldi.

L'Easy Script offre caratteristiche sofisticate quali:

• potenti aiuti nel produrre tabelle (definizione della larghezza dello schermo, tabulazione verticale ed orizzontale, tabulazione per i decimali);
• controlli di stampa completi (margini variabili, lunghezza variabile della pagina, sottolineature e evidenziazione);
• possibilità di facili aggiornamenti (posizionamento e sostituzione automatica, trasferimento di parte del testo, di parole, di frasi, di paragrafi, inserimenti, cancellazioni, annullamenti e fusione di testi);
• facile trattamento dei documenti, (collegamento a catena di più file, fusione automatica di testo ed indirizzi per la spedizione, ecc.).

L'Easy Script viene fornito in cartuccia e questo significa che è pronto immediatamente per l'uso. E' completo di un manuale di facile interpretazione che comprende una particolareggiata sezione d'istruzione. Ci sono esercizi di una difficoltà graduata che permettono a tutti di apprendere l'uso dell'Easy Script.

Attualmente è disponibile solo la versione inglese (Lire 85.000, Iva esclusa). Quella italiana è in preparazione.

Altri package

Gli altri pacchetti disponibili sono elencati di seguito. Tra parentesi il prezzo senza Iva.

• Linguaggio Forth 64 (95.000);
• Stat, statistiche (Lit. 95.000);
• DIARY, diario (Lit. 95.000);
• Calcresult in versione sia italiana che inglese espansa (Lit. 350.000) e semplificata (Lit. 175.000). ▪

Plottaggio in alta risoluzione

Un programma per tracciare grafici di funzioni in risoluzione 200 x 248 punti.

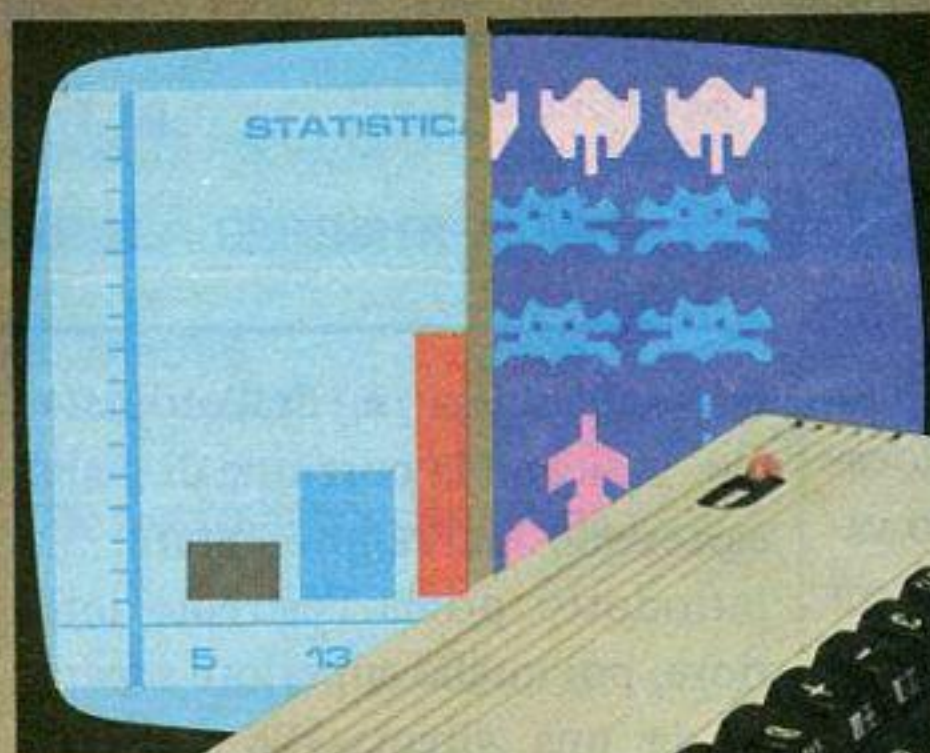
UNO degli aspetti più singolari dello schermo del Vic 20 è rappresentato dal bordo che circonda il riquadro ove vengono visualizzati i caratteri. In questo riquadro vi è spazio per 23 righe e 22 colonne, per una risoluzione di 176 x 182 punti.

Il seguente programma consente di "dilatare" lo schermo vero e proprio fino a fargli occupare tutto lo spazio utile del video televisivo. Ciò viene realizzato in due fasi: dapprima vengono modificati i registri che determinano il numero di righe e di colonne, che sono il 36866 e il 36867; successivamente viene corretta la "centratura" del riquadro usando i registri 36864 e 36865. Facendo poi partire la mappa dei caratteri da 7168, la mappa dei colori da 37188, le immagini dei caratteri da 5120 (cioè inserendo 253 nel registro 36869), e utilizzando alcune veloci subroutine in linguaggio macchina, è possibile tracciare grafici di funzioni in risoluzione 200 x 248 con risultati veramente apprezzabili.

Le subroutine sono in tutto tre. La prima azzera i 2K da 5120 a 7167, ossia le immagini dei caratteri, pone a 255 tutti i byte della mappa caratteri e a 5 i byte della mappa colori. Infine manipola tutti i registri suddetti (cioè quelli da 36864 a 36867). La seconda subroutine plotta un punto la cui ascissa è data dalla locazione 1 e la cui ordinata è data dalla locazione 2. Per origine degli assi è stato assunto l'angolo in basso a sinistra. Infine con la terza subroutine tutto ritorna come prima, l'immagine scompare e lo schermo può essere utilizzato normalmente.

Il funzionamento della seconda subroutine si divide in due fasi: la prima consiste nel calcolare, in base all'ascissa e all'ordinata, in quale locazione della mappa caratteri cade il punto da rappresentare. Successivamente viene

VIC 20
"COMPUTER DELL'ANNO"

L. 423.000

VIDEOGIOCHI
disponibili varie cassette videogame.

PERIFERICHE
si possono collegare direttamente registratore, dedicato, floppy, stampante, joystick.

TASTIERA
professionale con 66 tasti, 256 combinazioni di colori, 512 segni diversi per creare grafici, disegni, videogiochi.

SOFTWARE
una vasta gamma di programmi per tutte le necessità.

SUONO
3 toni e 9 ottave disponibili per creare le tue musiche ed effetti speciali.

VIC 20 è il computer più diffuso nel mondo, oltre un milione di esemplari venduti. Perchè?
1. VIC 20 è facile da usare: lo colleghi alla TV e premi i tasti. Tutto qui. - 2. Ci puoi giocare: dagli scacchi agli invasori spaziali, e puoi inventarti i tuoi giochi. 3. Lo usi per la scuola, per risolvere problemi e compiti. - 4. Lo impieghi in casa, per la contabilità entrate-uscite, ma anche per le ricette e per il bioritmo... 5. Costa solo 423.000 lire più iva. - 6. Puoi imparare a programmare in Basic, la lingua del futuro.
7. È stato scelto come "computer dell'anno" dalle migliori riviste internazionali di informatica.
8. È della Commodore, la N° 1 in Europa e in Italia nel campo dei Personal Computer.
Perchè non farti un regalo intelligente?

ETHOS PISA

individuato il gruppo di 8 byte che fornisce l'immagine del carattere puntato da suddetta locazione (facendo cioè 5120 + 8 x valore della locazione). Infine viene settato il X bit corrispondente al punto da plottare.

Come si è già detto le locazioni della mappa caratteri vengono inizializzate a 255; ciò vuol dire che se il programma trova che il punto da plottare cade in una locazione contenente 255, questo valore verrà modificato in maniera tale che punti ad 8 byte non ancora puntati da nessun altra locazione della mappa caratteri. A tale scopo il programma gestisce una variabile alla locazione 8191, che fornisce il valore da inserire nella suddetta locazione della mappa caratteri e che viene incrementata di uno ogni volta.

Dopo aver modificato il valore della locazione, il programma riprende da dove era stato interrotto. Si tratta di calcoli non semplici, la cui programmazione in Basic sarebbe stata più agevole, ma con tempi di esecuzione proibitivi.

Sono in Basic i due programmi, START e FUNZIONI, che hanno rispettivamente la funzione di caricare le tre subroutine, e di tracciare funzioni utilizzando le tre subroutine. In questo ultimo programma la funzione da plottare viene definita alla linea 20.

Il programma ci chiede se vogliamo modificarla e se rispondiamo "S" viene listata la linea 20 e il programma si ferma. Altrimenti va avanti richiedendo gli estremi dell'ascissa e dell'ordinata, che indicano la porzione di piano cartesiano da rappresentare. Va introdotto prima l'estremo inferiore, poi quello superiore separati da una virgola Se presenti, vengono tracciati anche i due assi. Se non compare nulla significa che il programma ha eseguito un calcolo impossibile, come divisione per zero o radice di un numero negativo ecc., o che la funzione non si trova in quella parte di piano.

Listato 1 - START

```
10 POKE51,21:POKE52,19:POKE55,21:POKE56,19:CLR
30 OPEN1,1,0,"HIRE"
40 FORK=4886TO5119
50 INPUT#1,A
60 POKEK,A
70 NEXTK
80 CLOSE1
```

Per far ripartire il programma basta premere la barra spaziatrice. Le tre subroutine, tranne la prima, sono completamente rilocabili. Va comunque ricordato che le locazioni 1 e 2 non contengono più, dopo che il punto è stato plottato, i due valori iniziali.

L'unico, vero inconveniente è che non tutto lo schermo può essere riempito: pur essendoci più di 700 caratteri visualizzabili, se ne vengono utilizzati più di 255, si avrà l'amara sorpresa di vedere lo schermo riempirsi di pallini o lineette in un indescrivibile caos. Ad ogni modo, questo inconveniente non dovrebbe verificarsi in un programma di plottaggio di funzioni monodrome, dove cioè ad ogni x corrisponde una e una sola y.

Il programma "funzioni"

Le prime linee di programma chiedono se si intende modificare la funzione da tracciare e provvedono al suo listaggio se la risposta è affermativa.

Le linee 30 E 40 richiedono gli estremi sull'ascissa e sull'ordinata e riformulano la richiesta se non viene inserito prima l'estremo inferiore e poi quello superiore.

Nella linea 50 e 60 si pongono le seguenti costanti: SX = scala delle ascisse, SY = scala delle ordinate e S = punto di partenza della seconda subroutine. Nella linea 60 viene lanciata la prima subroutine in linguaggio macchina.

La linea 70 dà inizio al ciclo FOR K/NEXTK che racchiude tutte le operazioni per il plottaggio vero e proprio.

Esso parte sfasato di un punto, dato che il programma unisce due punti con differenza di ascisse uguale a 1 attraverso il ciclo FOR J/NEXT J.

Le linee 100 e 110 tracciano l'asse delle ascisse, mentre le li-

Listato 2 - FUNZIONI

```
5 POKE36879,8:PRINT"ЭЖ"
8 PRINT"VUOI CAMBIARE LA ":PRINT"ЖFUNZIONE ";
10 INPUTA$:IFASC(A$)=83THENLIST20
20 DEFFNA(X)=SIN(X)
30 PRINT"ЖЖESTREMI DEL-":PRINT"ЖL' ASCISSA ";:INPUTA,B:IFB<=ATHENPRINT"TTTTT";:G
OTO30
40 PRINT"ЖESTREMI DEL-":PRINT"ЖL' ORDINATA ";:INPUTM,N:IFN<=MTHENPRINT"TTTT";:GO
TO40
50 SY=247/(N-M):SX=(B-A)/199:S=4886
60 PC=.5:P=INT((FNA(A)-M)*SY+PC):X=-1:U=1:D=2:SYS(5040)
70 FORK=A+SXTOB+SXSTEPSX:X=X+U
80 Y=INT((FNA(K)-M)*SY+PC):FORJ=PTOYSTEPSGN(Y-P)
85 IFP<0ORP>247THEN95
90 POKEU,X:POKED,J:SYS(S)
95 P=Y:NEXTJ,K
100 P=INT(-M*SY+PC):IFP<0ORP>247THEN120
110 FORK=0TO199:POKEU,K:POKED,P:SYS(S):NEXT
120 P=INT(-A/SX+PC):IFP<0ORP247THEN140
130 FORK=0TO247:POKEU,P:POKED,K:SYS(S):NEXT
140 GETA$:IFA$<>" "THEN140
150 PRINT"Э":SYS(5004):RUN
```

Listato 3 - STAMPA

```
2 OPEN1,4
5 CMD1
7 DEFFNA(X)=X+48-(X>9)*7
10 GOTO100
30 PRINT:PRINT"CODICI ESADECIMALI"
40 FORK=ATOBSTEP5
45 PRINTK;"  ";
50 FORJ=KTOK+4
55 IFJ>BTHENJ=K+5:GOTO80
60 S=PEEK(J):GOSUB1000
70 PRINTA$;"  ";
80 NEXTJ:PRINT:NEXTK
90 RETURN
100 PRINT"PRIMA SUBROUTINE"
110 A=5040:B=5119:GOSUB30:GOSUB2000
120 PRINT:PRINT"SECONDA SUBROUTINE"
130 A=4886:B=5003:GOSUB30:GOSUB2000
140 PRINT:PRINT"TERZA SUBROUTINE"
150 A=5004:B=5039:GOSUB30:GOSUB2000:END
1000 T=INT(S/16):R=S-T*16
1010 A$=CHR$(FNA(T))+CHR$(FNA(R)):RETURN
2000 PRINT:PRINT"CODICI DECIMALI"
2010 FORK=ATOB:PRINTPEEK(K),:NEXT:PRINT:RETURN
```

nee 120 e 130 tracciano quello delle ordinate.

Le ultime linee del programma creano un ciclo di attesa che si interrompe quando è premuta la barra spaziatrice.

Il programma può essere fermato semplicemente premendo il tasto di SHIFT destro e RUN STOP se il calcolatore è in attesa di dati (ossia sta eseguendo un'istruzione di INPUT), oppure RUN STOP e RESTORE se il calcolatore sta tracciando una funzione.

Una cosa da ricordare è che il programma START ridimensiona la memoria disponibile per programma e variabili a un solo Kbyte e che occorre spegnere il calcolatore per ritornare ad avere i 3,5 Kbyte della versione base. ▪

Paolo Savi
Via Kennedy, 18 20097 S. Donato (MI)

Gestione veloce delle cassette

E' NOTO che la ricerca veloce ed il caricamento in memoria di programmi registrati (in successione) su nastro magnetico per un loro eventuale aggiornamento e reinserimento non è possibile, salvo laboriose procedure e perdite di tempo. Le due brevi routine che proponiamo risolvono questo problema.

La prima routine va registrata all'inizio di ogni cassetta denominandola 'tempo richiesto'; la seconda va collocata all'inizio di ogni programma e serve per registrare nuovamente il programma eventualmente corretto. In più concorre a posizionare il nastro per la richiesta o la registrazione di nuovi programmi, nella stessa o in altre cassette, variando semplicemente il tempo 'HHMMSS'.

I listati delle routine sono illustrati nel riquadro, dove RET = RETURN.

I programmi da registrare vengono posizionati sul nastro in base al tempo prestabilito e che si legge nella riga 59, come spieghiamo in seguito.

Uso della routine

Ogni programma registrato ha un nome. All'esterno della cassetta, oltre ai nomi dei programmi, occorre annotare anche il tempo di posizionamento di detto programma, ad esempio:

LOAD 'tempo richiesto' 000000
LOAD 'settimana ' 000009
LOAD 'mese ' 000022
....
LOAD 'anno ' 000103.

All'accensione della macchina ed introdotta la cassetta, si premono i tasti SHIFT-RUN indi il tasto play del registratore. Verrà visualizzata la routine 'tempo richiesto'. A questa domanda, dopo aver riavvolto il nastro (REW) si batte il tempo del programma che si vuole richiamare, pari a quello precedentemente annotato sulla cassetta indi i tasti (RET.-F,FWD). Il nastro verrà posizionato all'altezza del programma desiderato. Ora basta battere NEW SHIFT-CLR indi i tasti SHIFT-RUN/PLAY ed il programma richiesto sarà caricato nella memoria del PET.

Se si deve modificare questo programma e quindi memorizzarlo di nuovo, basterà riavvolgere il nastro col tasto REW, battere 'RUN 4' sulla tastiera e poi premere 'RET.-F.FWD. Il nastro viene posizionato al punto giusto per registrarvi con la solita 'SAVE' il programma testè modificato.

Se si desidera richiamare un altro programma, dopo aver sempre riavvolto il nastro, occorre dapprima modificare il valore assegnato alla variabile TI$='0000XY' della riga 5, assegnandole il valore corrispondente al programma che vogliamo richiamare e che troviamo nella tabellina a suo tempo compilata.

Si dovrà battere quindi 'RUN 4' indi RET.-F.FWD, il nastro verrà posizionato all'altezza del programma desiderato. A questo punto si batte (NEW SHIFT-CLR) indi SHIFT-RUN - PLAY ed il programma verrà visualizzato sul monitor. ■

Marino Zambotti
Via Stelline, 9 20146 Milano

```
10 FM=59411: FO=249: PE=61:PRINT'S'
11 PRINT'RIAV.-REW- BATTI IL TEMPO RICHIESTO
12 INPUT'INDI RET.-F.FWD';W$:TI$='000000'
14 IF VAL(TI$)<VAL(W$) THEN 16
18 POKEFO,52:POKEFM,PE:PRINT'TEMPO RICHIESTO=';W$
20 PRINT'ORA BATTI -NEW- INDI (SHIF-RUN)':STOP
```

```
 2 RUN 20
 4 TI$='000000':PRINTTI$
 5 IF TI$='000030'THENPOKE249,52:POKE59411,61:PRINT TI$:STOP
 6 GOTO 5
30.........(PROGRAMMA)
```

IMPARA A PROGRAMMARE CON IL VIC

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

★ DISPENSA N. 4 ★

Commodore è vicina.

Napoli Parco S. Paolo

Commodore aggiunge un nuovo punto vendita e assistenza ai quasi 500 già operanti in Italia:
CM Computermarket,
Napoli Parco S. Paolo, Is. 9
Tel. 081/76.72.222.
Qui trovi gente che la sa lunga sui computer e sulle qualità dei vari computer; professionisti che ti propongono sistemi con il miglior rapporto prezzo-prestazioni e una vasta gamma di soluzioni: gamma alla quale nessun produttore di "personal" si può oggi avvicinare. Anche per questo Commodore è prima in Italia. E in Europa.
Ti aspettiamo.

ETHOS PISA

6. poke, peek e il colore

Codici carattere

Come avete già visto all'inizio di queste lezioni, nel computer i caratteri e i numeri vengono memorizzati sotto forma di configurazioni binarie. I codici binari standard sono stati fissati da diverse organizzazioni internazionali, ma quelli più largamente adottati sono i codici definiti dalla American Standard Code for Information Interchange (ASCII) che riporteremo in appendice così come risultano implementati sul VIC. I caratteri possono essere convertiti in questi codici e viceversa con la funzione stringa ASC e CHR$, descritte qui di seguito.

• *La funzione CHR$.*

Questa funzione dà il carattere corrispondente ad uno specifico codice ASCII, cioè:

```
10 LET A$ = CHR$(66)
```

Il codice ASCII per la lettera B è 66, quindi l'istruzione memorizza B in A$. Le parole vengono formate mediante concatenazione, vale a dire che

```
10 LET A$ = CHR$(66)+ CHR$(69)
```

dà per risultato A$ = BE, dove 69 è il codice ASCII corrispondente alla lettera E.

Notate che poichè la funzione CHR$ dà il codice ASCII, le variabili possono essere usate, se necessario, per diversi caratteri di controllo (per esempio, il RETURN).

• *La funzione ASC*

E' l'opposto della funzione CHR$, in quanto fornisce il numero di codice ASCII per un dato carattere. Ad esempio

```
10 LET X = ASC("E")
```

dà come risultato X = 69.

Se l'argomento è una variabile stringa, viene fornito il codice ASCII del primo carattere, vale a dire che, ad esempio,

```
5 LET T$ = "TOTAL"
10 LET X  = ASC(T$)
```

danno per risultato X = 84.

• *LEN.*

La funzione LEN fornisce la lunghezza di una stringa. Cambiando, ad esempio, la linea 10 in modo da avere

```
10 LET L = LEN(T$)
```

si avrebbe come risultato L = 5.

• *Le funzioni LEFT$ e RIGHT$*

Le due funzioni individuano il numero specificato di caratteri più a sinistra o più a destra di una data stringa. Così, ad esempio,

```
10 LET B$ = LEFT$(T$,2)
```

fornisce i due caratteri più a sinistra della stringa T$, cioè B$= TO, e

```
10 LET E$ = RIGHT$(T$,3)
```

dà per risultato E$ = TAL.

• *La funzione MID$*

La funzione MID$ calcola una sottostringa di un carattere a partire dall'iesimo carattere, cosicchè ad esempio,

```
10 LET C$ =   MID$(T$,2,3)
```

dà per risultato C$ = OTA dove n = 3 e i = 2.

• *La funzione STR$*

La funzione STR$ converte un argomento numerico in una stringa equivalente della sua forma stampata.

```
10 LET N = 20
20 LET X$ = STR$(N)
```

ad esempio, dà per risultato X$ contenente "20" come stringa, perciò

```
30 LET Y$ = "VIC" + X$
40 PRINT Y$
```

dà come risultato la stampa dell'espressione

```
VIC 20
```

Ricordate che la versione alfanumerica del valore numerico contiene uno spazio iniziale (il segno + che viene eliminato).

• *La funzione VAL*

La funzione VAL costituisce l'esatto contrario della funzione STR$: essa infatti esamina la stringa da sinistra verso destra fornendo il primo formato numerico riconoscibile. Ne consegue che

X=VAL("VIC 20") dà per risultato

```
X =  20
```

e

X =VAL("-78.97.65") dà per risultato

```
X = -78.97
```

POKE E PEEK

Il comando POKE permette di porre qualsiasi valore specificato direttamente in una data locazione di memoria. La sua forma generale è la seguente:

```
POKE x,y
```

dove il valore di x specifica la locazione di memoria e y il valore che deve essere elaborato dal comando POKE.

```
POKE 36879,93
```

ad esempio, pone il valore 93 nella locazione di memoria 36879, che in questo caso determina il colore dello schermo, per cui il valore 93 fà in modo che lo schermo diventi completamente verde.

Per quanto riguarda il VIC, l'uso più comune del comando POKE è quello di controllo del colore, del suono e della posizione dei caratteri sullo schermo. Alcune tecniche di programmazione speciali permettono di utilizzare i comandi PEEK e POKE per eseguire operazioni dirette ad una speciale area della memoria del computer, nota come pagina zero, per poter ottenere effetti speciali non realizzabili con il BASIC. Benchè l'uso dei due comandi non possa in alcun modo danneggiare il vostro VIC, prima di cimentarvi con le soluzioni più sofisticate, fate in modo di acquisire una buona pratica nelle applicazioni più semplici.

Se in qualsiasi momento il VIC sembra smettere di funzionare correttamente, ad esempio scompare il cursore, premete contemporaneamente i tasti STOP e RESTORE in modo da riportare il VIC al punto di partenza senza perdere il programma.

La funzione PEEK permette l'accesso al contenuto di una specifica locazione di memoria, quindi PEEK (x) dà il valore del contenuto della locazione x, che può essere assegnata come variabile nel solito modo, cioè

```
10 LET S = PEEK(36879)
20 PRINT S
```

Eseguendo questo programma subito dopo l'accensione del VIC, S dovrebbe avere il valore 27.

La funzione PEEK può essere usata nei giochi e nelle animazioni per determinare lo stato delle varie parti dello schermo, per esempio, per sapere se un missile occupa al momento la posizione corrispondente all'obiettivo, il chè significa un punto a favore del giocatore.

Sia PEEK che POKE accettano la variabili quindi tutti gli esempi che seguono sono sintatticamente validi:

```
POKE X,Z
POKE X + L*22,Z
PEEK (X + L*22)
IF A$ = CHR$(PEEK(K)) THEN Y
```

Il controllo del colore del VIC

Il sistema di colore del VIC permette il controllo indipendente del colore dello schermo, della sua cornice e da tutti i caratteri destinati ad essere visualizzati.

La cornice può essere di otto colori diversi, lo

schermo di 16, tutti impostabili con un solo comando POKE diretto alla locazione 36879, il cui valore determina la combinazione di colore schermo/cornice. In appendice riporteremo i valori di tutte le combinazioni possibili. La combinazione all'accensione, ad esempio, è blu con schermo bianco ottenibile anche tutte le volte che si effettua un ripristino con il comando

POKE 36879,27

Il programma nella tabella 6.1 permette di ottenere una dimostrazione di tutte le combinazioni cornice/schermo possibili. Ricordate comunque che quando i caratteri sono dello stesso colore dello schermo, non sono più leggibili e apparentemente scompaiono Il programma è costituito da due loop: quello esterno (dalla linea 10 alla linea 60) imposta S su valori di schermo successivi associati ad una cornice nera e da qui al primo ingresso di ciascuna linea dell'appendice; quello interno (dalla linea 20 alla linea 50) invece determina la posizione di colonna e quindi S + B rappresenta il valore da elaborare con il comando POKE.

La linea 40 è un ritardante che consente di osservare ciascuna combinazione di colori.

La modifica e la regolazione del colore possono essere effettuate in diversi modi, come vedremo in seguito.

```
10 FOR S=8 TO 248 STEP 16
20 FOR B=0 TO 7
30 POKE 36879,S+B
40 FOR D=1 TO 1000: NEXT D
50 NEXT B
60 NEXT S
70 END
80 :
90 :
```

Tabella 6.1. Programma per la riproduzione di più combinazioni di colori per cornice e schermo.

Il controllo del colore dei caratteri

Il colore dei caratteri può essere programmato mediante tasti specifici, attribuendo delle variabili a particolari codici di carattere o utilizzando il comando POKE.

Il colore dei caratteri visualizzati a partire dalla posizione corrente del cursore può essere selezionato battendo il tasto CTRL contemporaneamente ad uno dei tasti che contrassegnano il colore. La stessa sequenza può essere chiusa fra virgolette in un'istruzione PRINT, cioè nel modo seguente:

10 PRINT "CTRL e il tasto Rosso THIS PRINTS RED"

Come alternativa, è possibile assegnare alla sequenza una variabile stringa da incorporare in un'istruzione PRINT, come nelle istruzioni

10 C$ = "CTRL e il tasto Rosso"
20 PRINT C$: "THIS PRINTS RED"

Un terzo modo per ottenere la stampa di caratteri in un certo colore predefinito consiste nell'utilizzare l'apposito codice di carattere, il cui elenco è riportato in appendice. Il programma nella tabella 6.2 offre un esempio di questo metodo in cui i codici di carattere vengono letti da una serie di istruzioni DATA. Il metodo di modifica del colore caratteri è particolarmente utile nella stampa dei titoli e delle intestazioni.

Se usata con i tasti RVS ON e RVS OFF anche la barra spaziatrice permette di ottenere delle barre di diverso colore. Infatti, ogni volta che viene usato il tasto RVS ON per l'attivazione del set di caratteri in negativo, lo spazio si trasforma in un blocco

```
10 FOR L=1 TO 8
20 READ C,C$
30 PRINT CHR$(C);"QUESTO E' ";C$
40 NEXT L
100 DATA 144,NERO
110 DATA 5,BIANCO
120 DATA 28,ROSSO
130 DATA 159,AZZURRO
140 DATA 156,VIOLA
150 DATA 30,VERDE
160 DATA 31,BLU
170 DATA 158,GIALLO
180 END
190 :
200 :
```

Tabella 6.2 Stampa di stringhe in più colori

intero del colore che in quel momento risulta selezionato. Oltre che essere eseguita da tastiera, anche in questo caso la sequenza può essere programmata, come nell'esempio seguente:

10 PRINT "CTRL e RVS ON e CTRL e tasto Rosso "

dove l'istruzione permette di ottenere una barra rossa lunga cinque spazi. Notate che entrambi i comandi RVS ON e RVS OFF hanno dei codici di carattere (per la precisione 18 il primo e 146 il secondo) utilizzabili nei programmi nello stesso modo in cui vengono usati i codici per i colori. Quindi il precedente esempio potrebbe risultare così:

```
10 PRINT CHR$(18);CHR$(28);
```

Il quarto metodo possibile per ottenere caratteri colorati è quello di usare il comando POKE descritto nel prossimo paragrafo.

Colorare i caratteri con il comando POKE

Lo schermo del video è mappato in memoria. In altre parole, ogni possibile posizione di carattere dello schermo ha una propria relativa locazione di memoria. Nel VIC, nella sua configurazione base, la posizione dell'angolo in alto a sinistra, ad esempio, corrisponde alla locazione di memoria 7680 e, usando il comando POKE a questo indirizzo con un particolare valore, è possibile ottenere la comparsa di un certo carattere di colore bianco (e in questo caso sarà opportuno che per lo schermo sia stato selezionato un colore contrastante). Vi ricordiamo, a questo proposito, che i codici di schermo di tutti i tasti, che verranno riportati in appendice, non corrispondono ai codici di carattere e sono rilevanti solo per l'uso del comando POKE di visualizzazione. Quindi.

```
POKE 7680,83
```

produce un piccolo cuore in corrispondenza della posizione di riposo del cursore. Se ora aggiungete 128 al codice, otterrete lo stesso carattere visualizzato in negativo.

Per ciascuna locazione di memoria dello schermo esiste una corrispondente locazione di memoria del codice colore; perciò 38400 è la locazione di colore corrispondente alla posizione di schermo 7680. Ricordate che è importante che vi accertiate che esista l'esatta corrispondenza tra locazione di colore e locazione di memoria dello schermo.

Il colore del carattere depositato in una data locazione può essere modificato inserendo un valore pari al numero indicato su ciascun tasto di colore meno 1, ad esempio 2 per il colore rosso. In questo modo

```
POKE 7680.83 : POKE 38400,2
```

consente di collocare un cuore rosso nella posizione di riposo del cursore.

Anche le locazioni di memoria dello schermo assieme alle relative locazioni di colore saranno riportate in appendice e valgono per il VIC nella sua configurazione base. Se a quest'ultima si aggiungono altri 3K, tutte le locazioni subiscono un cambiamento. Per questa ragione, durante la stesura di programmi che implicano l'uso dei comandi PEEK e POKE in funzione della visualizzazione sullo schermo, è consigliabile usare espressioni che siano in relazione con la posizione di riposo del cursore e con le combinazioni e i valori richiesti.

L'espressione, inoltre, può usare la riga e la colonna richieste per calcolare la locazione di memoria come nell'esempio seguente:

```
100 SM=7680: CM=38400
110 POKE SM+22*R+C, 83
120 POKE CM+22*R+C, 2
130 :
140 :
```

Se avete un VIC con più di 3K di espansione RAM, avrete bisogno di porre SM a 4096 e CM a 37888.

La routine riportata qui sopra posiziona un cuore rosso sulla riga R della colonna C. Se la memoria dello schermo e le locazioni di colore cambiano, sarà sufficiente modificare solo la linea 100.

La tabella 6.3 offre un esempio di programma molto semplice che chiarisce l'uso del comando POKE, che in questo caso inserisce una serie di cuori di diverso colore quasi al centro dello schermo.

La posizione di partenza è rappresentata dalla

riga 11 (linea 110), segue quindi un loop (dalla linea 120 alla linea 150) che stampa un cuore, ogni volta di un colore diverso, nelle colonne C + I (dalla colonna 6 alla colonna 13 comprese). Il programma fa inoltre uso di un contatore di loop (I) che permette di fissare il colore I-1 nella linea 140.

Un esempio di istogramma

Il programma riportato nella tabella 6.4 dimostra come è possibile usare i colori per produrre un istogramma. La routine produce la traccia su video di dieci linee verticali di colore rosse in una scala da 0 a 10.

La linea 10 azzera lo schermo; la 20 stampa l'intestazione principale; le linee dalla numero 30 alla numero 60 costituiscono un loop che crea la scala verticale. Il trattamento delle stringhe ottenuto con queste linee determina il corretto allineamento dei numeri ottenuti con il conteggio del loop. Il carattere grafico della linea 50 rappresenta una parte dell'asse verticale in scala, mentre la linea 70 determina l'asse orizzontale.

In questo esempio, i dati sono contenuti in una linea DATA (160) e vengono letti ogni volta che è necessario per ciascuna colonna della linea 90 nella variabile D.

L'istogramma viene prodotto inserendo con il comando POKE dei caratteri negativi di colore rosso verticalmente sulle dieci colonne previste.

Le linee dalla numero 80 alla numero 150 costituiscono un loop esterno che ripete il loop interno

```
90 PRINT"♥": REM CANCELLAZIONE SCHERMO
100 SM=7680: CM=38400:REM VIC NON ESPANSI
105 REM SOSTITUIRE SM=4096: CM=37888 NEL
106 REM CASO DI VIC DOTATI
107 REM DI ESPANSIONI DI MEMORIA
110 R=11: C=6
120 FOR I=1 TO 8
130 POKE SM+22*R+C+I, 83
140 POKE CM+22*R+C+I, I-1
150 NEXT I
160 END
170 :
180 :
```

Tabella 6.3 Inserimento di cuori colorati con il comando POKE

```
10 PRINT"♥": REM CANCELLAZIONE SCHERMO
20 PRINT"*****ISTOGRAMMI*****"
25 PRINT
30 FOR L=10 TO 1 STEP -1
40 LS$="   "+STR$(L)
50 PRINT RIGHT$(LS$,4);"┐":REM TASTO SHIFT + P
60 NEXT L
70 PRINT"    0┬┬┬┬┬┬┬┬┬┬"
80 FOR P=1 TO 10
90 READ D
100 IF D=0 THEN 150
110 FOR H=1 TO D
120 POKE 7970-(H*22)+P, 160:REM VIC NON ESPANSI
130 POKE 38690-(H*22)+P, 2:REM VIC NON ESPANSI
131 REM SOSTITUIRE POKE 4386-(H*22)+P, 160
135 REM POKE 38178-(H*22)+P, 2
136 REM NEL CASO DI VIC DOTATI
137 REM DI ESPANSIONI DI MEMORIA
140 NEXT H
150 NEXT P
160 DATA 5,7,8,3,5,2,6,0,1,6
170 END
180 :
190 :
```

Tabella 6.4 Routine per la realizzazione di un istogramma

per ciascuna colonna. Se l'istogramma ha un ingresso zero, il loop interno viene superato senza essere eseguito per effetto della linea 100.

Il loop interno (dalla linea 10 alla linea 140) inserisce un carattere spazio negativo (32+128 = 160) verticalmente D volte.

Animazioni

La formazione delle barre nell'istogramma dell'esempio precedente rappresenta un tipo molto semplice di animazione. Per poter avere però forme più elaborate come quelle usate, ad esempio, nei giochi, è innanzitutto indispensabile poter avere movimenti in sù e in giù su tutta la superficie dello schermo.

Per poter avere l'illusione del movimento, occorre spostare un certo carattere da una locazione dello schermo ad un'altra opportunamente ravvicinata e sostituire la locazione precedentemente occupata con un carattere spazio. Tutto ciò può essere fatto con il comando POKE.

Il programma nella tabella 6.5 rappresenta un facile esempio di animazione in cui un asterisco viene spostato lungo tutto lo schermo in senso orizzontale da sinistra verso destra lungo la riga 8 (notate che la linea in alto è identificata come linea zero). Le linee dalla numero 10 alla numero 30 fissano la visualizzazione iniziale e le locaaioni di colore alla prima posizione della riga 8. Il loop (dalla linea 40 alla linea 90) sposta il carattere richiesto su ciascuna colonna di quella riga colorandolo di rosso (linea 60). La linea 70, all'interno del loop, inserisce uno spazio nella colonna precedente, mentre il loop ritardante nella linea 80 impedisce all'asterisco di spostarsi troppo rapidamente. La linea 100, infine, inserisce uno spazio nella posizione finale dell'asterisco all'uscita dal loop.

```
5 PRINT"▯": REM CANCELLAZIONE
10 SM = 7680: CM = 38400
20 S = SM+22*8
30 C = CM+22*8
40 FOR I=0 TO 21
50 POKE S+I, 42
60 POKE C+I, 2
70 POKE S+I-1, 32
80 FOR D=1 TO 100: NEXT D
90 NEXT I
100 POKE S+21, 32
110 END
120 :
130 :
```

Tabella 6.5 Esempio di semplice animazione

Problema

Usando i caratteri grafici, disegnate sullo schermo un semplice viso stilizzato ed introducete l'animazione in modo che il viso appaia alternativamente imbronciato e sorridente.

Un esempio della routine che vi serve in questo caso è riportato nella tabella A9. ▪

Come recuperare i file persi

Questo programma permette di recuperare i file persi accidentalmente sui dischetti da 5" 1/4. E' adatto per i vari drive (8050, 4040, 3040) e computer della Commodore (serie 2001, 3000, 4000 e 8000) ed è adattabile al Vic.

IL COMANDO SCRATCH (nelle tre versioni: come comando diretto nel Basic 4.0 o attraverso il canale comandi 15 o col comando diretto >$ del Dos Support) non cancella fisicamente sul dischetto il file a cui viene applicato, ma agisce solo sulla directory a sulla BAM, rendendo libero a successive scritture i blocchi che il file occupava.

La directory è l'elenco dei file presenti sul dischetto, ed occupa quasi tutti i settori della traccia 39 per il drive (unità a dischi) 8050 (traccia 18 per il 4040 e simili). Ogni settore contiene i puntatori (traccia, settore) al prossimo settore della Directory (0,0 se è l'ultimo) e la descrizione di otto file (entrata). Ogni entrata assume l'aspetto di fig. 1.

La BAM (Block Available Map = mappa dei blocchi disponibili), è contenuta nei settori 0 e 3 della traccia 38 per l'8050 (settore 0, traccia 18 per il 4040) e ci dice quali settori sono disponibili e quali no, attraverso una mappa di 4 byte per ogni traccia.

Il comando SCRATCH altera,

Formato della directory
(traccia 18, settore 1 per 3040, 4040)
(traccia 39, settore 1 per 8050)

Byte	Contenuto
0,1	Traccia, settore del prossimo blocco della directory
2-31	Entrata del file 1
34-63	Entrata del file 2
66-95	Entrata del file 3
98-127	Entrata del file 4
130-159	Entrata del file 5
162-191	Entrata del file 6
194-223	Entrata del file 7
226-255	Entrata del file 8

Struttura di una singola entrata

Byte	Contenuto
0	128+tipo 'OR' del tipo con $80 per indicare un file chiuso correttamente tipo 0: cancellato 1: sequenziale 2: programma 3: user (utente) 4: relativo
1,2	Traccia, settore del primo blocco di dati
3-18	Nome del file completato da spazi in shift
19,20	Solo per i file relativi: traccia,settore
21	Dimensione record
22-25	Non usati
26,27	TR.,Set. quando si sostituisce un file
28,29	Numero di blocchi nel file parte bassa, parte alta

Fig. 1: struttura della directory e di un'entrata di file.

RECUPERO FILE

```
100 PRINT"■ ■.RECUPERO FILE ■■"
110 REM EDOARDO PATRUCCO   1982
120 REM MICRO & PERSONAL COMPUTER
130 REM ===========================
140 DIMFE$(32),TF$(5):X$="":R$=""
150 FORI=0TO5:READTF$(I):NEXT:CC=0
160 K=0:CH=9:S=1:TR=39:T=TR:DR$="0"
170 IN$="■TIPO   NOME FILE      TR.SET.LUNG."
180 PRINT"1 RECUPERO FILE CANCELLATO■"
190 PRINT"2 RECUPERO DI TUTTI I FILE CANCELLATI■"
200 PRINT"3 CONVALIDA■"
210 PRINT"4 VISUALIZZAZIONE  DIRECTORY AMPLIATA■"
220 PRINT"5 CANCELLAZIONE FILE■":PRINT"6 FINE■"
230 PRINT"<SPACE> PER TORNARE AL MENU■"
240 PRINT"■1:REC1;2:RECT;3:CONV;4:DIR;5:CANC;6:END■";
250 GETR$:IFR$=""THEN250
260 IFR$<"1"ORR$>"6"THEN250
270 R=VAL(R$):PRINTR$:RT=0
280 ONRGOTO450,960,1000,1500,1800,2000
290 END:REM ============================
295 REM ===== LEGGE,STAMPA DIRECTORY =
298 REM ============================
300 TT=ASC(FE$(1)):S8=ASC(FE$(2))
310 FT=ASC(FE$(0))AND127
320 IFFT>5THENPRINT"TIPO ERRATO":FT=5
330 TF$=TF$(FT)
340 LF=ASC(FE$(28))+256*ASC(FE$(29))
350 TT$=RIGHT$("  "+STR$(TT),3)
360 SS$=RIGHT$("  "+STR$(SS),3)
370 LF$=RIGHT$("      "+STR$(LF),5)
380 PRINT"■"TF$"■ "NF$;TAB(20);
390 IFSPTHENPRINT#4,"■"TF$"■ "NF$;
400 PRINTTT$;SS$;LF$
410 IFSPTHENPRINT#4,TT$;SS$;LF$
420 RETURN
440 REM ============================
450 PRINT"■RICERCA DI UN CERTO FILE"
455 REM ============================
460 GOSUB750
470 PRINT"NOME DEL FILE DA RECUPERARE"
480 INPUTNO$:L=LEN(NO$):FORK=1TOL
490 IFMID$(NO$,K,1)="*"THENL=K-1:K=80
500 NEXTK:IFK<80THENL=16
510 NO$=LEFT$(NO$+"                ",L)
520 PRINT"STO CERCANDO    ■"NO$
525 REM ============================
530 REM ==RICERCA FILE NELLA DIRECT.==
535 REM ============================
540 T=TR:OPEN1,8,15:PRINT#1,"I"+DR$
550 S=1:OPENCH,8,CH,"#"+DR$
560 PRINT#1,"U1:"CH;DR;T;S
570 PRINT#1,"B-P:"CH;0:GET#CH,T$,S$
```

```
580 FORFE=1TO8
590 GETR$:IFR$=" "THEN710
600 FORCC=0TO31:GET#CH,X$
610 IFX$=""THENX$=CHR$(0)
620 FE$(CC)=X$:NEXT:NF$=""
630 FORK=3TO18:NF$=NF$+FE$(K):NEXT
640 IFNO$=LEFT$(NF$,L)THEN1200
650 NEXTFE
660 IFT$=""THENT$=CHR$(0)
670 IFS$=""THENS$=CHR$(0)
680 T=ASC(T$):S=ASC(S$)
690 IFT<>0THEN560
700 PRINT"▒NON TROVATO■"NF$
710 CLOSECH:CLOSE1:CLOSE4:GOTO240
715 REM ============================
720 REM = GESTIONE ERRORI DA DISCO =
725 REM ============================
730 INPUT#1,CE$,ER$,ET$.ES$:IFCE$="00"THENRETURN
740 PRINTCE$,ER$,ET$;ES$:GOTO710
745 REM ============================
750 REM ====== QUALE DRIVE ? =======
755 REM ============================
760 PRINT"DRIVE      ";DR$"■■■";:INPUTDR$
770 IFDR$<>"0"ANDDR$<>"1"THEN760
780 DR=VAL(DR$)
790 RETURN
795 REM ============================
800 REM === SCELTA TIPO DI FILE ====
805 REM ============================
810 PRINT"▒0=DEL;1=SEQ;2=PRG;3=USR;4=REL;5=>MENU"
820 INPUT"▒TIPO■    2■■■";R:IFR<0ORR>5THEN820
830 IFR=5THEN710
840 PO=2+32*(FE-1):TP=R+128
850 PRINT#1,"U1:"CH;DR;T;S
860 PRINT#1,"B-P:"CH;PO
870 PRINT#CH,CHR$(TP);
880 PRINT#1,"U2:"CH;DR;T;S
890 RETURN
895 REM ============================
900 REM == IL FILE VA RECUPERATO ? =
905 REM ============================
910 IFMID$(NF$,1,1)=CHR$(0)THEN1420
920 INPUT"▒DA RECUPERARE (S/N/END)■    N■■■";R$
930 IFR$="S"THENGOSUB800:PRINT#1,"B-P:"CH;PO+32
940 IFLEFT$(R$,1)="E"THEN1420
950 RETURN
960 REM =======================
970 PRINT"▒RECUPERO TUTTI I FILE"
980 RT=-1:GOTO1510
990 REM =======================
1000 PRINT"▒CONVALIDA ■";:GOSUB750
1010 CLOSE1:OPEN1,8,15:PRINT#1,"V"+DR$
1020 GOSUB720:GOTO710
1200 REM =====================
```

```
1210 PRINT"TROVATO IL FILE "NF$
1220 PRINTIN$
1230 GOSUB300
1330 PRINT"CHE TIPO ERA  ?"
1340 GOSUB800
1420 INPUT"CONVALIDA (S/N)    S███";R$
1430 IFR$="S"THEN1000
1440 GOTO710
1450 REM ==============================
1500 PRINT"♥VISUALIZZAZIONE/STAMPA  DIRECTORY"
1505 REM ==============================
1510 GOSUB750:INPUT"STAMPA    N███";R$
1520 SP=0:IFR$="S"THENSP=-1:OPEN4,4
1530 GOSUB1900:PRINTIN$:S=1
1540 IFSPTHENPRINT#4,IN$
1550 IFSTTHENPRINT"STAMPANTE SPENTA":END
1560 OPEN1,8,15:PRINT#1,"I"+DR$
1570 OPENCH,8,CH,"#"+DR$:GOSUB720
1580 PRINT#1,"U1:"CH;DR;T;S
1590 PRINT#1,"B-P:"CH;0:GET#CH,T$,S$
1600 FORFE=1TO8
1610 GETR$:IFR$=" "THEN1730
1620 FORCC=0TO31:GET#CH,X$
1630 IFX$=""THENX$=CHR$(0)
1640 FE$(CC)=X$:NEXT:NF$=""
1650 FORK=3TO18:NF$=NF$+FE$(K):NEXT
1660 GOSUB300:IFRT AND(FT=0)THENGOSUB900
1670 IFFE$(3)=CHR$(0)THEN1730
1680 NEXTFE
1690 IFT$=""THENT$=CHR$(0)
1700 IFS$=""THENS$=CHR$(0)
1710 T=ASC(T$):S=ASC(S$)
1720 IFT<>0THEN1580
1730 IF RT THEN1420
1740 T=TR:GOTO710
1790 REM ==============================
1800 PRINT"♥CANCELLAZIONE FILE":GOSUB750
1810 PRINT"NOME DEL FILE DA CANCELLARE"
1820 INPUTR$:CLOSE1:OPEN1,8,15
1830 PRINT#1,"S"+DR$+":"+R$
1840 GOSUB720:GOTO710
1890 REM ==============================
1900 REM LEGGE NOME,IDENTIF.DISCHETTO
1910 PO=6:IFTR=18THENPO=144
1920 S=0:OPEN1,8,15,"I"+DR$:GOSUB720
1930 OPENCH,8,CH,"#":PRINT#1,"U1:"CH;DR;TR;S
1940 PRINT#1,"B-P:";CH;PO:NF$=""
1950 FORK=1TO20:GET#CH,X$:NF$=NF$+X$:NEXT
1960 PRINTDR$"   "NF$:IFSPTHENPRINT#4,DR$"   "NF$
1970 CLOSECH:CLOSE1:RETURN
1980 DATADEL,SEQ,PRG,USR,REL,XXX
1990 REM ==============================
2000 PRINT"♥FINE":CLOSE1:CLOSECH:CLOSE4:END
READY.
```

nell'entrata del file, il tipo sostituendo a quello preesistente il tipo "Deleted" (cancello), contrassegnato dal valore 0. Inoltre rende liberi nella BAM i blocchi. Lascia però inalterati il nome del file, il puntatore (traccia, settore) al suo primo blocco, la lunghezza in blocchi ed altre informazioni (che riguardano solo i file relativi).

L'unica informazione persa durante la SCRATCH, dunque, è quella sul tipo del file; esso viene perciò richiesto all'utilizzatore del programma (normalmente si propone il tipo 2 = tipo programma).

Questo programma permette di cercare nella directory, per nome, un determinato file che si era cancellato per errore, oppure di cercare sequenzialmente nella directory tutti i file, anche quelli cancellati, visualizzarli, chiedendo per ciascuno di quelli cancellati se si vuole salvarli.

Per la ricerca di un file con un certo nome vale la sintassi Commodore: se si imposta nomefile* si cerca il primo file il cui nome inizia con i caratteri nomefile.

Dopo aver eseguito queste operazioni ed aver quindi rintracciato la posizione nella directory dell'entrata del file, è necessario eseguire una convalida del dischetto. Ciò (comando VALIDATE o >V nel DOS Support o V attraverso il canale comandi 15) fa sì che risultino nuovamente occupati i blocchi deallocati dallo SCRATCH, cosicchè, se si scrivono nuovi file, questi non vengano sovrapposti.

Ovviamente il programma va eseguito prima di aver fatto qualsiasi operazione di scrittura o copiatura sui settori che il file da recùperare occupava.

La convalida (che oltre a riallocare i blocchi ricostruisce i collegamenti tra i vari settori del file) non è automatica, ma viene comunque ricordata all'utente quando necessaria. Ciò perchè per dischetti molti pieni può durare alcuni minuti, e si vuole permettere all'utente di abortire quanto fatto.

Uso del programma

Risulta agevole: è prevista la stampa della "directory ampliata", che dice anche quali sono i file cancellati e traccia e settore del primo blocco di ogni file.

Sapere dove inizia fisicamente sul dischetto un certo file può essere molto utile in certi casi. Ad esempio, poichè il terzo e quarto

Elenco variabili usate

CC:	Contatore generico get da disco
CE$:	Codice errore
CH:	Canale del disco : posto=9
DR:	Drive unità a disco: vale 0 0 1
DR$:	Analogo
ER$,ES$,ET$:	Errori da disco
FE:	Indice del for per i file entry
FE$<	Vettore di 32 elementi: contiene i caratteri di ogni entrata di file
FI:	Tipo del file: 0-4
I,K:	Contatori generici
IN$:	Intestazione (per il menù)
L :	Lunghezza di NO$
LF,LF$:	Lunghezza nome file
NF$:	Nome del file letto da disco: 16 car.
NO$:	Nome del file che cerco
PO:	Puntatore accesso diretto
R,R$:	Input generici
RT:	Segnalatore ricerca tutti i file se = 0, se no li stampa
S,T:	Traccia,settore directory
SS,TT:	Traccia,settore di un file
SS$,TT$:	Analoghi
TR:	Traccia della directory = 39 0 18
SP:	Segnalatore: <> 0: stampa, se no no
X$:	Input generico
TF$:	Tipo file
TF$<:	Vettore: del, seq, prg, usr, rel, xxx

Fig. 2: elenco delle variabili

byte del primo blocco di ogni file di tipo programma dicono da dove va memorizzato il file nella memoria del calcolatore, alterando con pochi comandi di accesso diretto al disco questi puntatori si rialloca tutto il programma senza doverlo leggere come file con lunghi cicli for...next. Questa tecnica risulta utile per programmi in linguaggio macchina che non abbiano salti a indirizzi assoluti al proprio interno: rende inoltre possibile disassemblare da RAM un programma copiato su dischetto da una ROM non in proprio possesso. Sono possibilità riservate agli esperti, su dischetto poco pieni, al fine di limitare il rischio di perdere dati importanti per aver commesso qualche errore di calcolo dei puntatori.

Modifiche e note

I DRIVE con 35 tracce (3040/4040/etc.) richiedono TR=18 nella linea 160.

Questo programma non è stato volutamente scritto in linguaggio macchina per essere più facilmente modificabile e/o ampliabile e per poter girare su macchine delle varie serie Commodore, che hanno le routine di gestione dei dischi differenti ed allocate a indirizzi diversi su ROM. E' comunque abbastanza veloce: la ricerca (lineare) dell'ultimo file (il 224°) su un dischetto dell'8050 richiede 2 minuti e 1/2. D'altra parte un numero così elevato di file si ha solo su dischetti-archivio, che non si cancellano mai. Ciò che richiede molto tempo è invece la convalida del dischetto, che non si può accelerare, essendo eseguita dal sistema, nè interrompe. Eventuali comandi inviati all'unità a disco durante la convalida bloccano il sistema fino al termine della convalida.

E' consigliabile lavorare su dischetti con pochi file: si hanno ricerche di file più veloci, directory più leggibili e convalide più rapide, facilitando così l'eventuale uso di questo programma, da tenere come rimedio nel malaugurato caso che si cancelli accidentalmente un file che serviva ancora. Naturalmente i file cancellati con calamite non sono recuperabili con questo programma. ▪

Edoardo Patrucco
Via Alessandria, 181
San Germano Casale Monf. (AL)
Tel. (0142) 50343

```
C R O S S     R E F E R E N C E

PROGRAMMA : RECUPERO FILE

------*-------------------------- - - -
VAR.  ! LINEA DEL PROGRAMMA
------*-------------------------- - - -
      !
CC    !  150   600   620  1620  1640
CE$   !  730   740
CH    !  160   550   560   570   600   710   850   860   870   880   930
      ! 1570  1580  1590  1620  1930  1940  1950  1970  2000
DR    !  560   780   850   880  1580  1930
DR$   !  160   540   550   760   770   780  1010  1560  1570  1830  1920
      ! 1960
ER$   !  730   740
ES$   !  730   740
ET$   !  730   740
FE    !  580   650   840  1600  1680
FE$(  !  140   300   310   340   620   630  1640  1650  1670
FT    !  310   320   330  1660
I     !  150
IN$   !  170  1220  1530  1540
K     !  160   480   490   500   630  1650  1950
L     !  480   490   500   510   640
LF    !  340   370
LF$   !  370   400   410
NF$   !  380   390   620   630   640   700   910  1210  1640  1650  1940
      ! 1950  1960
NO$   !  480   490   510   520   640
PO    !  840   860   930  1910  1940
R     !  270   280   820   830   840
R$    !  140   250   260   270   590   920   930   940  1420  1430  1510
      ! 1520  1610  1820  1830
RT    !  270   980  1660  1730
S     !  160   550   560   680   850   880  1530  1580  1710  1920  1930
S$    !  570   670   680  1590  1700  1710
SP    !  390   410  1520  1540  1960
SS    !  300   360
SS$   !  360   400   410
ST    ! 1550
T     !  160   540   560   680   690   850   880  1580  1710  1720  1740
T$    !  570   660   680  1590  1690  1710
TF$   !  330   380   390
TF$(  !  140   150   330
TP    !  840   870
TR    !  160   540  1740  1910  1930
TT    !  300   350
TT$   !  350   400   410
X$    !  140   600   610   620  1620  1630  1640  1950
```

fig. 3 : elenco delle variabili

Ecco dove trovi i Personal Computer Commodore

Distributori Commodore

Liguria
Pirisi Informatica
Piazza Cavour, 19 - 16043 Chiavari
Tel. 0185/30.10.31

Piemonte
Aba Elettronica di Caramia
Via Fossati, 5/C - 10141 Torino
Tel. 011/33.20.65

Lombardia
Homic Personal Computers srl
Piazza de Angeli, 3 - 20146 Milano
Tel. 02/49.88.201

Veneto, Friuli-Venezia Giulia, Trentino-Alto Adige
CO.R.EL. Friuli Computers
Via Mercatovecchio, 28 - 33100 Udine
Tel. 0432/29.14.66

Emilia-Romagna, Marche
S.H.R. srl
Via Faentina 175/A
48010 Fornace Zarattini (Ravenna)
Tel. 0544/46.32.00

Toscana
M.C.S. Spa
Via Pier Capponi, 87 - 50132 Firenze
Tel. 055/57.13.80

Umbria - Alto Lazio
Atlas System srl
Via Guglielmo Marconi, 17 - 01100 Viterbo
Tel. 0761/22.46.88

Lazio
Kiber Italia srl
P.le Asia, 21 - 00144 Roma Eur
Tel. 06/59.16.438

Abruzzo
Pragma System srl
Via Tiburtina, 57 - 65100 Pescara
Tel. 085/50.883

Campania
Computer Market
Parco S. Paolo Isolato, 9 - 80100 Napoli
Tel. 081/76.72.222

Puglia
Maselli per l'ufficio
Via L. Zuppetta, 5 - 71100 Foggia
Tel. 0881/76.111

Puglia
Business Automation System
Largo De Gemmis, 46/B-46/C-48-48/A-48/B
70124 Bari - Tel. 080/22.75.75-22.73.44

Calabria
Sirangelo Computers srl
Via Nicola Parisio, 25 - 87100 Cosenza
Tel. 0984/75.741

Sicilia
Edilcomput Progetti dell'Ing. Giuseppe Carbone
Via La Farina, 141 Is. L
98100 Messina
Tel. 090/29.28.269

Sardegna
S.I.I. - Sistemi Integrati Informatica
Via S. Lucifero, 95 - 09100 Cagliari
Tel. 070/66.37.46

Ecco dove trovi VIC-20

Distributori Commodore

Negozi Expert

Negozi G.B.C.

Negozi Singer

Temporex Italiana

Salmoiraghi

La Rinascente

Bit Shop Primavera

Alessandria Via Savonarola, 13
Ancona Via De Gasperi, 40
Arezzo Via Filippo Lippi, 13
Barletta Via Vitrani, 58
Bari Via Capruzzi, 192
Bassano del Grappa Via Jacopo Da Ponte, 51
Bergamo Via F. D'Assisi, 5
Biella Via Italia, 50/A
Bologna Via Brugnoli, 1
Cagliari Via Zagabria, 47
Campobasso Via Mons. II Bologna, 10
Cesano Maderno Via Ferrini, 6
Cinisello Balsamo V.le Matteotti, 66
Como Via L. Sacco, 3
Cosenza Via Dei Mille 86
Cuneo Corso Nizza, 16
Favria Canavese C.so Matteotti, 13
Firenze Via G. Milanesi, 28/30
Foggia Via Marchianò, 1
Forlì P.zza M. Degli Ambrogi, 1
Gallarate Via A. Da Brescia, 2
Genova Via D. Fiasella, 51/r
Genova-Sestri Via Chiaravagna, 10/R
Imperia Via Delbecchi, 32
L'Aquila Via Strada 85,2
Lecco Via L. Da Vinci, 7
Livorno Via San Simone, 31
Lucca Via San Concordio, 160
Macerata Via Spalato, 126
Merano Via Santa Maria del Conforto, 22
Messina Via Del Vespro, 71
Milano Via Jacopo Palma, 9
Milano Viale Certosa, 91
Milano Via Petrella, 6
Milano Via G. Cantoni, 7
Milano P.zza Firenze, 4
Milano Via Altaguardia, 2
Milano V.le Corsica, 14
Mirano-Venezia Via Gramsci, 40
Monza Via Azzone Visconti, 39
Morbegno Via Fabani, 31
Napoli Corso Vittorio Emanuele, 54
Napoli Via Luigia Sanfelice, 7/A
Novara Baluardo Q. Sella, 32
Padova Via Fistomba, 8
Palermo Via Libertà, 4
Parma Via Imbriani, 41
Pavia Via C. Battisti, 4/A
Perugia Via R. D'Andreotto, 49/55
Pescara Via Tiburtina, 264/bis
Pescara Via Trieste, 73
Piacenza Via IV Novembre, 60
Pisa Via XXIV Maggio, 101
Pistoia Via Adua, 350
Potenza Via Mazzini, 72
Pozzuoli Via Pergolesi, 13
Rimini Via Bertola, 75
Roma Via Ponzio Cominio, 46
Roma Via Cerreto Da Spoleto, 23
Roma P.zza San Donà Di Piave, 14
Roma V.le Quattro Venti, 152
Roma Largo Belloni, 4
Savona Via G. Scarpa, 13/r
Sondrio Via Nazario Sauro, 28
Teramo Via Martiri Pennesi, 14
Terni Via Beccaria, 20
Torino Via Chivasso, 11
Torino C.so Grosseto, 209
Torino Via Tripoli, 179
Trento Via Sighele, 7/1
Treviglio Viale Buonarroti, 5/a
Trieste Via F. Severo, 138
Udine Via Tavagnacco, 89/91
Varese Via Carrobbio, 13
Verona Via Pontiere, 2
Viareggio Via Volta, 79
Voghera P.zza Carducci, 11

commodore COMPUTER

Giornale luminoso

Come fare un listino animato per la vetrina del negozio.

LA routine che segue realizza sul video un effetto molto simpatico, per cui può essere inserita in altri programmi oppure utilizzata a scopo pubblicitario nei negozi. Essa, infatti, simula lo scroll orizzontale di una scritta inserita precedentemente.

Le poche righe di programma non sono di difficile comprensione e quindi la loro spiegazione si può riassumere in questo REMark.

10) Inserimento della frase da far scorrere sullo schermo (massimo 83 caratteri) e pulitura schermo.

20) Vengono aggiunti 22 spazi alla stringa inserita in linea 10 in modo che la scritta compaia da destra.

30) Ciclo interdipendente con la stampa della frase. Il ciclo è fino a 128 in quanto la scritta deve scomparire nella 1° colonna. Sono necessari quindi altri 23 cicli, oltre ai primi 83, per permettere la scomparsa della scritta sul video.

40) Porta il cursore in alto a sinistra e scende di 10 righe, dopo di che stampa i 22 caratteri più a destra partendo dalla posizione mobile I-22.

```
10 INPUT A$:PRINT CHR$
(147)
20 A$="                    "+A$
30 FOR I=23 TO 128
40 PRINT CHR$(19);TAB(220)
;MID$(A$,I-22,22);" "
50 FOR J=I TO 100:NEXTJ,I:
GOTO 30
```

50) Ciclo a vuoto per la regolazione della velocità dello scroll, chiusura del ciclo principale e ripristino dello stesso al suo termine.

Per ottenere 10 scritte identiche che scorrono sullo schermo invece è sufficiente sostituire la linea 40 in:

40 FOR K=1 TO 10:PRINT MID $ (A $,I-22,22);" ": NEXTK :PRINT CHR$(19).

Con leggere modifiche la routine può essere adattata anche ai fratelli maggiori della famiglia Commodore. ■

Corrado Montalti
Via Lidice, 4
40139 Bologna

COMMODORE
ASSISTENZA TECNICA
Hardware
Contratti di manutenzione
e riparazione di
tutti i sistemi
Commodore dal
super pet al
Vic 20
Novità microfazer
Buffer esterno da
8K e 64 K per
eliminare i tempi morti
delle stampe
Software
Exteded basic
level II
Oltre 50 comandi
in più a disposizione
del vostro
computer
Compilatore,
per rendere bidirezionale
la stampante 4022

Caratteri grafici

L'8032, soprattutto per chi possedeva il 4032, ha un handicap: non consente l'uso dei caratteri grafici direttamente dalla tastiera. Questo inconveniente viene superato con la routine che vi proponiamo.

QUANTE volte vi è capitato di avere bisogno di un carattere grafico e quante difficoltà avete incontrato per ottenerlo non essendo disponibile direttamente dalla tastiera dell'8032?

Nell'8032 la Commodore ha eliminato l'accesso dalla tastiera a molti caratteri grafici, perciò, chi volesse ancora usarli, deve fare uso dei CHR$ che ne complicano notevolmente l'uso.

Le routine che seguono servono ad ovviare questo problema creando un tasto di controllo che ha la funzione dello SHIFT dei computers Commodore precedenti.

L'uso è molto semplice; infatti, una volta fatto partire il programma, è sufficiente premere il tasto 2", che funge da tasto di controllo e, tenendolo abbassato, premere un qualsiasi altro tasto, anche se shiftato, e sul video comparirà il carattere grafico corrispondente a questo.

Premendo il tasto 2" sul video comparirà un 2 lampeggiante indicante che il programma è in funzione.

Qualcuno si sarà chiesto come mai ci sono 3 programmi dei quali 2 in Basic. La risposta è semplice: infatti ho cercato di dare la possibilità di caricare il programma direttamente come istruzione in linguaggio macchina o per mezzo del basic; inoltre ho cercato di risolvere il problema per coloro che nelle locazioni dal 666 al 785 hanno un programma in linguaggio macchina, dando la possibilità di allocare il programma in qualsiasi zona di memoria compresa tra 1030 e 32647.

Il secondo programma carica la routine in L.M. dall'indirizzo a1, perciò è sufficiente alla riga 240 porre un valore ad a1 per scegliere l'inizio della routine in L.M.

Lasciando la riga 240 intatta, il programma legge il limite massimo della memoria RAM per il basic e calcola il valore di a1 affinchè la routine si collochi nella zona più alta della memoria RAM.

Nella scelta del valore di a1 bisogna fare alcune considerazioni, infatti il programma n. 2 limita automaticamente la memoria usata dal basic ad a1, che comunque non deve essere minore di 1024 nè maggiore di 32647. Ciò spiega la presenza di altri 2 programmi dei quali uno è in basic ed ha la funzione di caricare la routine in L.M. passando per il basic.

Il programma in L.M. inizia dalla locazione 666 ovvero 029A in esadecimale nel buffer della prima cassetta.

Il programma in L.M., una volta caricato, va inizializzato con SYS 677, mentre per abilitarlo occorre SYS 666 e per disa-

Programma n.ro 1 in basic

```
100 print"s" tab(18) chr$(15);
110 print" ###########################################
120 print" #                                         #
130 print" #     CARATTERI GRAFICI PER CBM 8032      #
140 print" #                                         #
150 print" #                  BY                     #
160 print" #                                         #
170 print" #         Silvano  Bastianello            #
180 print" #                                  VS 1.0#
190 print" ###########################################
200 rem
210 print
220 print
230 print" Per abilitarlo...................SYS q 666 "
240 print
250 print" Per disabilitarlo................SYS q 777 "
260 print
270 print" Per  ottenere i caratteri  grafici  pigiare
280 print" il  tasto <2> e contemporaneamente il tasto
290 print" corrispondente al simbolo voluto.
300 fory=666to785
310 readx
320 pokey,x
330 next
340 print
350 print" o.k. puoi continuare"
360 print"sscccccccccccccccccccccccsys 666qqq"
370 new
380 end
390 data162,2,134,145,162,176,134,144,96,234,234,162,3
400 data133,42,232,133,43,76,154,2,234,165,151,201,50,240
410 data6,201,255,208,17,240,12,198,198,201,255,208,12
420 data169,32,164,198,145,196,141,191,2,76,85,228,169
430 data213,133,144,76,85,228,165,151,201,50,240,26,201
440 data255,208,25,230,198,165,170,201,1,208,10,164,198
450 data136,177,196,56,233,128,145,196,169,176,133,144,76
460 data85,228,164,198,136,177,196,24,105,64,162,50,142
470 data191,2,76,239,2,234,169,85,133,144,169,228,133,145
480 data96
```

bilitarlo si ricorre a SYS 777.

Una volta caricati, i programmi si fanno partire con RUN (conviene registrare il programma all'inizio del disco, infatti per caricarlo e farlo partire basta premere SHIFT-RUN/STOP).

Con il programma in basic n. 1 sotto READY, comparirà SYS 666, perciò se volete abilitarlo premete RETURN oppure SHIFT-RETURN.

Nel programma basic n. 2 la routine in L.M. viene abilitata automaticamente.

Per usare tutti i caratteri grafici consiglio di premere i due SHIFT e il tasto 2".

Per tornare ai caratteri minuscoli occorre usare **PRINT** CHR$(142).

I programmi si basano sul fatto che ad ogni carattere ne corrisponde uno di grafico il cui codice è uguale al codice del carattere incrementato di 64.

Nella tabella a corredo sono

Annuario
MPUTER
il "NEWSMAGAZINE" dell'INFORMATICA
.56-lire 5500
arche
nodelli
miliardi
milioni
di lire
In omaggio
il libro
"Come
scegliere un
omputer"

Programma n.ro 2 in basic

```
#########################################
#   CARATTERI GRAFICI PER CBM 8032      #
#                                       #
#                  BY                   #
#                                       #
#        Silvano  Bastianello           #
#                                 VS 2.0#
#########################################
100 print"▮" tab(18) chr$(15)
190 print"
200 print"
210 rem
220 a2=peek(52)+peek(53)*256
230 a1=a2-120
240 a1=a1 :rem numero della prima locazione del programma
250 forj=a1toa1+109
260 readx
270 ifx>0then330
280 i=a1-x-1
290 i1=int(i/256)
300 pokej,i-i1*256:j=j+1
310 pokej,i1
320 goto430
330 ifx<256goto420
340 i=a1+x-256
350 i1=int(i/256)
360 pokej,i1:j=j+1
370 pokej,133:j=j+1
380 pokej,145:j=j+1
390 pokej,169:j=j+1
400 pokej,i-i1*256
410 goto430
420 pokej,x
430 next
440 a2=int(a1/256):poke53,a2
450 poke52,a1-a2*256
460 print
470 print
480 print" Per abilitarlo....................SYS ▮";a1
490 print
500 print" Per disabilitarlo.................SYS ▮";a1+101
510 print
520 print" Per  ottenere i caratteri  grafici  pigiare
530 print" il  tasto <2> e contemporaneamente il tasto
540 print" corrispondente al simbolo voluto.
550 print"▮▮"
560 new
570 sys(a1)
580 data169,265,133,144,96,165,151,201,50,240,6,201,255
590 data208,17,240,12,198,198,201,255,208,12,169,32,164
600 data198,145,196,141,-25,76,85,228,169,306,133,144,76
610 data85,228,165,151,201,50,240,25,201,255,208,24,230
620 data198,165,170,201,1,208,9,164,198,136,177,196,56
630 data233,128,145,196,32,-1,76,85,228,164,198,136,177
640 data196,24,105,64,162,50,142,-25,76,-77,234,169,85
650 data133,144,169,228,133,145,96
```

indicati i tasti e i loro simboli corrispondenti. I numeri riportati sono i valori dei codici in ASCII cioè quello che si ottengono in ASC(...).

Praticamente il programma legge il codice del carattere nella memoria RAM video e lo aumenta di 64. Questa funzione viene svolta dalle locazioni 02F8-0304. Le istruzioni situate nelle locazioni 029A-02A2 e 0309-0311 modificano il vettore IRQ.

Il tasto funzione viene gestito nelle locazioni 02A7-02F7, che controllano se viene premuto il tasto 2" (02A7-02B5), se subito dopo viene digitato un altro tasto (02B0-02DE), se una volta lasciato il tasto 2" sul video il numero 2 non compaia in reverse (02DE–02F7) e se dopo il carattere grafico non venga scritto un 2 (02B6-02CA).

Parliamo ora delle locazioni, usate in questo programma, situate in pagina zero. Dal listato compaiono subito le locazioni n. 90-91 dove è situato il vettore IRQ.

Le locazioni 2A-2B contengono il puntatore di inizio variabili, mentre la AA contiene un valore uguale a uno se il cursore è in reverse.

Le locazioni maggiormente usate sono le C4-C5-C6. Nelle C4-C5 c'è l'indirizzo della prima locazione della riga video dove si trova il cursore.

Nella locazione C6 si trova il numero di caratteri che dista il cursore dall'inizio riga. ▪

Silvano Bastianello
Via Settimo, 49 - Tel. 0444/555551
36023 Longare (VI)

Programma in Linguaggio Macchina

```
.:  029a a2 02 86 91 a2 b0 86 90
.:  02a2 60 ea ea a2 03 85 2a e8
.:  02aa 85 2b 4c 9a 02 ea a5 97
.:  02b2 c9 32 f0 06 c9 ff d0 11
.:  02ba f0 0c c6 c6 c9 ff d0 0c
.:  02c2 a9 20 a4 c6 91 c4 8d bf
.:  02ca 02 4c 55 e4 a9 d5 85 90
.:  02d2 4c 55 e4 a5 97 c9 32 f0
.:  02da 1a c9 ff d0 19 e6 c6 a5
.:  02e2 aa c9 01 d0 0a a4 c6 88
.:  02ea b1 c4 38 e9 80 91 c4 a9
.:  02f2 b0 85 90 4c 55 e4 a4 c6
.:  02fa 88 b1 c4 18 69 40 a2 32
.:  0302 8e bf 02 4c ef 02 ea a9
.:  030a 55 85 90 a9 e4 85 91 60
```

Corrispondenza dei simboli

	32	=	160	@	64	= —	192
!	33	= ▌	161	A	65	= ♠	193
"	34	=	32	B	66	= |	194
#	35	= ‾	163	C	67	= —	195
$	36	= _	164	D	68	= —	196
%	37	= C	67	E	69	= ‾	197
&	38	= ▒	166	F	70	= _	198
'	39	= |	167	G	71	= |	199
(	40	= B	66	H	72	= |	200
)	41	= ◤	169	I	73	= ╮	201
*	42	= |	170	J	74	= ╰	202
+	43	= ├	171	K	75	= ╯	203
,	44	= ▖	172	L	76	= L	204
-	45	= └	173	M	77	= ╲	205
.	46	= ┐	174	N	78	= ╱	206
/	47	= ▂	175	O	79	= ┌	207
0	48	= ┌	176	P	80	= ┐	208
1	49	= ┴	177	Q	81	= ┌	207
2	50	= ┬	178	R	82	= _	210
3	51	= ┤	179	S	83	= ♥	211
4	52	= ▏	180	T	84	= |	212
5	53	= ▍	181	U	85	= ╭	213
6	54	= ▐	182	V	86	= ╳	214
7	55	= ‾	183	W	87	= ○	215
8	56	= ▀	184	X	88	= ♣	216
9	57	= ▃	185	Y	89	= |	217
:	58	= ┘	186	Z	90	= ♦	218
;	59	= ▗	187	[	91	= ┼	219
<	60	= ▝	188	\	92	= ▒	220
=	61	= ┘	189	]	93	= |	221
>	62	= ▘	190	↑	94	= π	255
?	63	= ▚	191	←	95	= ◥	223

L'aritmetica binaria

IL NUMERO 1789 è ritenuto, per la maggior parte della gente, è composto da un migliaio, sette centinaia, otto decine, nove unità, ovvero $1 \times 10^3, + 7 \times 10^2 + 8 \times 10^1 + 9 \times 10^0$. Ma cambiando la definizione del numero che costituisce la base, esso può significare:
$1 \times 16^3 + 7 \times 16^2 + 8 \times 16^1 + 9 \times 16^0$

Questo sistema di numerazione è detto esedecimale ed è la forma usata nel microprocessore.

Per distinguere tra numeri con differente base, i matematici usano scrivere 1789_{10} o semplicemente 1789 per i numeri in base dieci o decimali, e 1789_{16} per i numeri in base 16 o esadecimali. Poichè pochissimi computer o dispositivi I/O permettono di scrivere dei suffissi, tutti i numeri esadecimali sono preceduti da un segno $. Quindi 1789 significa in base dieci e $1789 significa in base sedici. Perchè si usa l'esadecimale? Perchè questo è il modo più conveniente per rappresentare due cifre con 8 bits.

Il 6500 usato nei sistemi Commodore è un microprocessore orientato al byte, il che significa che la maggior parte delle operazioni richiedono numeri binari da 8 bit.

Ci sono vari modi di guardare ad una serie di 8 bit. Il primo è di vederli come 8 bit separati, così che 0000 1000 significa che il bit N. 3 (considerando i bit numerati da 0 a 7) è attivo e che tutti gli altri sono azzerati. Si può anche considerare la stringa come un numero binario da 8 bit, nel qual caso il suo valore è:
$0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8$

Il numero può anche significare $ 08.

Per i fini dell'analisi logica, ciascun bit è considerato individualmente, ma per scopi aritmetici, gli 8 bit sono trattati come un numero binario.

Le regole principali dell'aritmetica binaria sono:

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 0$ con riporto di 1

Il riporto si ha se il numero risultante è maggiore della base come nel decimale 8 + 4 = 2 + 10.

In esadecimale; $8 + $4 = $C così che 8+ 4 ha un riporto in base 10 ma non in base 16. Usando queste regole per sommare 8 + 2 in binario, si ha quanto segue:

0000 1000	8	1×2^3
0000 0010	+2	1×2^1
0000 1010	10	$1 \times 2^3 + 1 \times 2^1$

Quindi ogni numero tra 0 e 255 può essere rappresentato con 8 bits, e la somma binaria può essere fatta usando le equazioni base dell'addizione binaria, $R_j = (A_j - B_j - C_{j-1})$ dove, come definito in precedenza, - è la notazione per l'OR esclusivo.

Nella maggior parte delle applicazioni è anche necessario sottrarre. Le operazioni di sottrazione richiedono un differente circuito hardware oppure un diverso modo di rappresentare i numeri.

Una combinazione dei due modi è di prevedere un semplice invertitore logico per ciascun bit. Questo produrrà:

0000 1100	12
1111 0011	-12

Ma, sottraendo 12 da 12, il risultato dovrebbe essere 0.

```
0000 1100    +12
1111 0011    -12
-------------------
1111 1111      0
```

Però, se viene aggiunto un riporto al numero complementato:

```
            1    riporto
0000 1100        12
1111 0011       -12
-----------------------
0000 0000         0
```

Se, invece di rappresentare -12 come complemento di 12, esso viene rappresentato come il complemento più il riporto, si ottiene quanto segue:

```
                  —
1111 0011        12
        1    riporto
-------------------
1111 0100       -12
0000 1100       +12
-------------------
0000 0000         0
```

Questa rappresentazione viene chiamata "complemento a 2" e rappresenta il modo nel quale sono trattati i numeri negativi nel microcalcolatore.

La rappresentazione esadecimale è la rappresentazione dei numeri in base 16.

Siccome 16 è un multiplo di 2, l'esadecimale è una abbreviazione utile per la rappresentazione di quattro cifre binarie o bit, mantenendo le regole aritmetiche.

```
   Binario     esadec.
  0100 1111       4F
+ 0110 0010     + 62
----------------------
  1011 0001       B1
```

Per trarre vantaggio da questa abbreviazione, tutti gli indirizzi in questo testo sono mostrati in notazione esadecimale. Si deve notare che il lettore dovrà imparare ad operare in esadecimale il più presto possibile. La continua traduzione in decimale costituisce sia perdita di tempo che fonte di errori. Il lavorare in esadecimale ed in binario ridurrà il tempo di apprendimento della manipolazione in esadecimale e la familiarità con l'uso di questa utile rappresentazione.

Per quanto molte applicazioni del microcomputer possono essere fatte con successo con le operazioni binarie, alcune operazioni sono meglio eseguite in decimale. Per quanto l'uso di un carattere decimale per byte possa essere una via legittima per risolvere il problema, questo costituisce un uso incompleto della capacità di un byte da 8 bit.

Il microprocessore permette l'uso della rappresentazione BCD contratta (packed), che usa 4 bit per cifra. Ad esempio, in BCD il numero 79 è rappresentato:

Binario		BCD		Esadec.
0111 1001	=	79	=	79

Il microprocessore tiene automaticamente conto di ciò e si corregge per il fatto che:

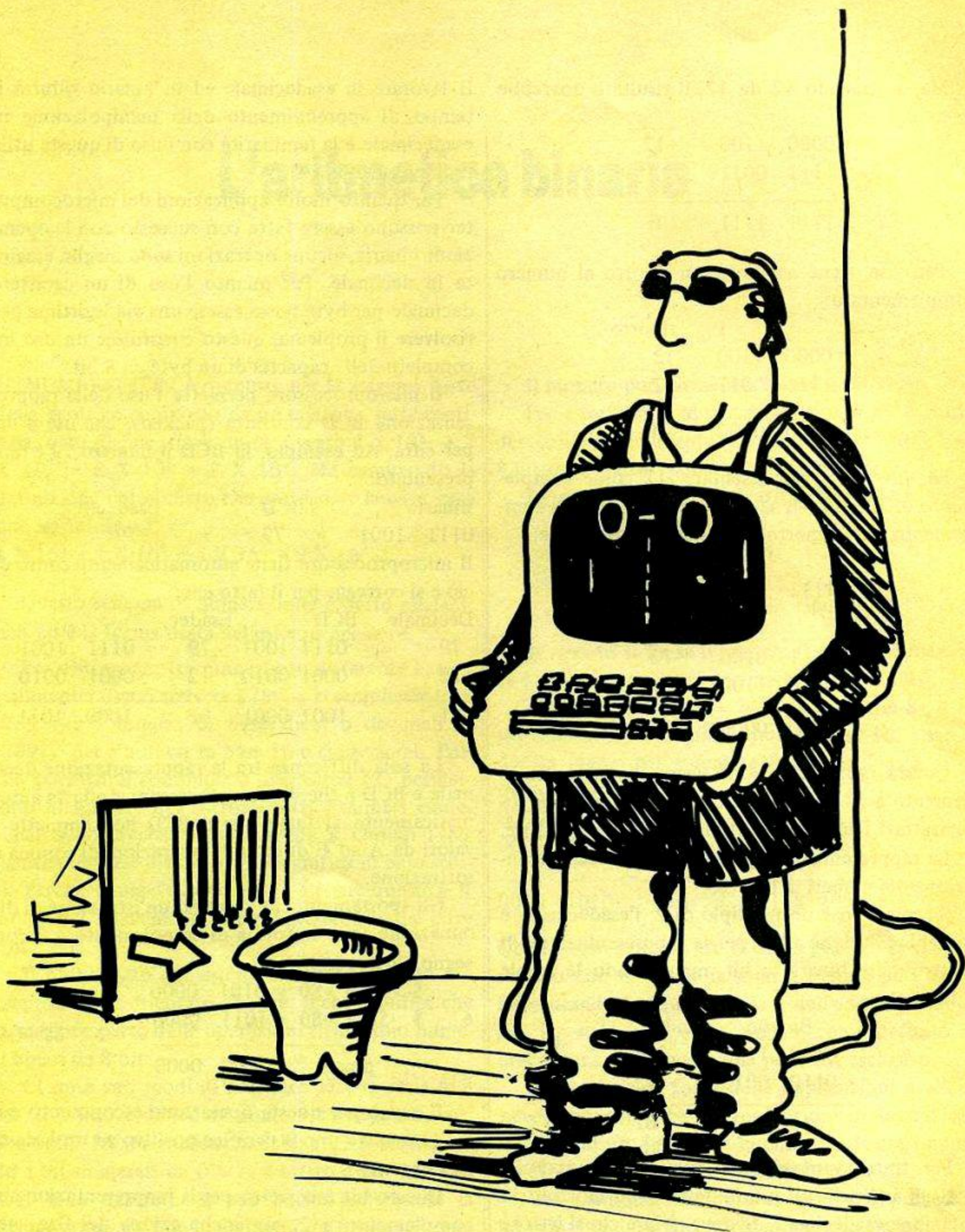
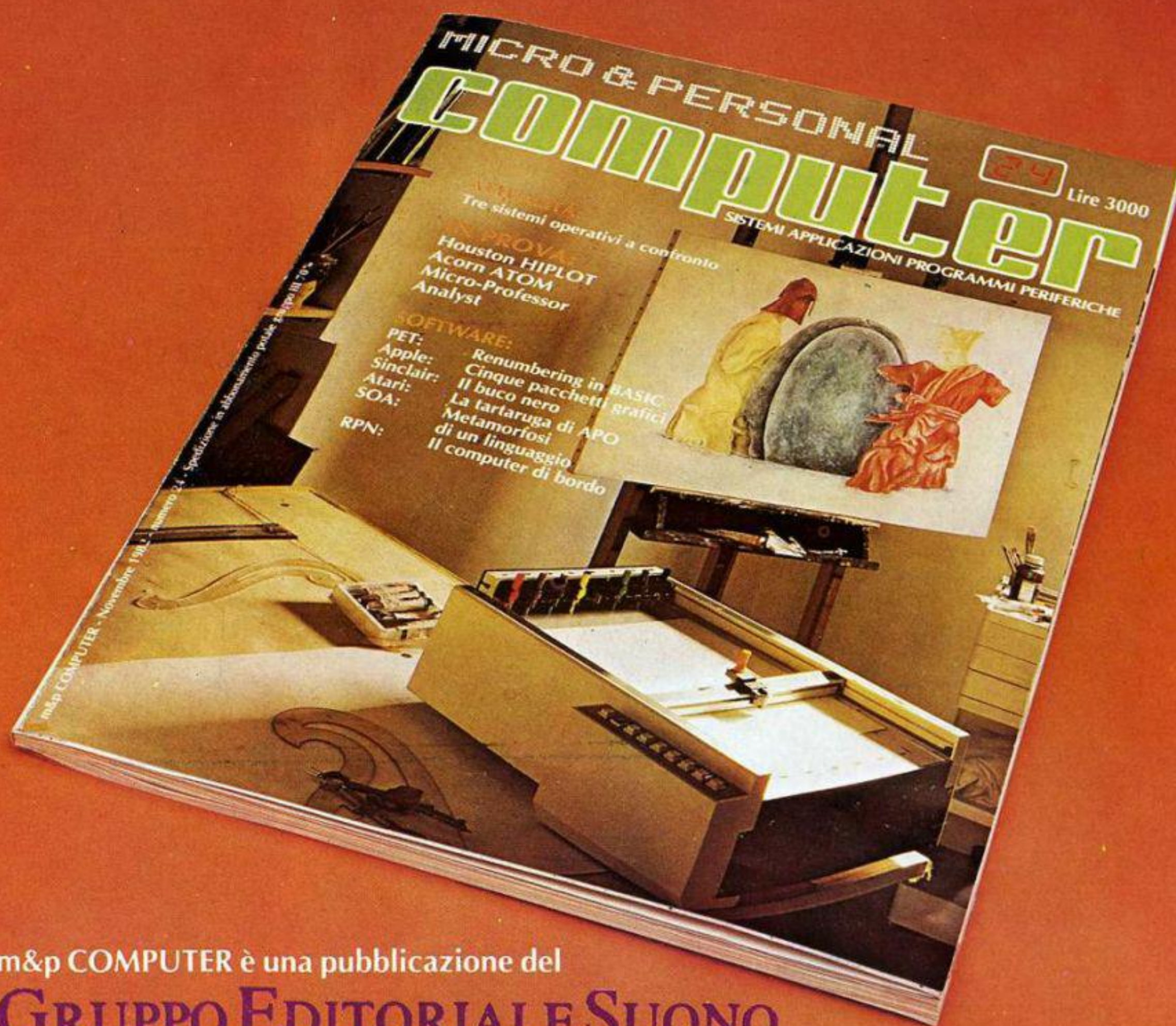
Decimale	BCD	Esadec.		
79	0111 1001	79	=	0111 1001
+12	0001 0010	12		0001 0010
91	1001 0001	8B		1000 1011

La sola differenza tra la rappresentazione decimale e BCD è che il microprocessore si adatta automaticamente al fatto che il BCD non ammette i valori da A ad F durante le operazioni di somma e sottrazione.

Gli spostamenti che seguono un'istruzione di diramazione sono in forma di complemento a 2 con segno, il che significa:

```
   $ +50 = +80 = 0101 0000
e  $ -50 = -80 = 1011 0000
                 ---------
       prova     0000 0000
```

Il segno per questa operazione è contenuto nel bit N.7, dove uno 0 significa positivo ed un 1 significa negativo.

Questo bit è corretto per la rappresentazione in complemento a 2, ma anche aziona dei flags nel microprocessore che indicano se sia o no il caso di riportare o di prendere un resto dal byte di indirizzo alto. ▪

ENTRA NELLA NUOVA DIMENSIONE... LEGGI

m&p COMPUTER è una pubblicazione del

GRUPPO EDITORIALE SUONO

Via del Casaletto 380 - 00151 Roma